Distributed Information Processing in Neural-Inspired Microelectronic Circuits

Dissertation*

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Federico Corradi

aus

Italien

Promotionskomitee

Prof. Dr. Giacomo Indiveri (Vorsitz)

Prof. Dr. Tobi Delbruck

Prof. Dr. Rodney Douglas

Zürich, 2015

GUT ZUM DRUCK

U. Grossmann

Der Studiendekan

Datum: 21.9.15

Contents

1 Introduction **1**

1.1 Background 1

2 Neural information processing in physical systems **4**

2.1 Cortical computation and canonical macro circuits 4

2.2 Large-scale spike-based computer models 6

2.3 Large-scale neuromorphic hardware systems 7

2.4 Reconfigurable hardware for spiking-neural networks 9

2.5 Discussion: the right question for the appropriate scale of investigation . . 10

3 Electronic neurons and synapses **12**

3.1 Motivations: Mimicking neurobiology with analog neuromorphic Very Large Scale Integration (VLSI) circuits 12

3.2 Neurons 14

3.2.1 Model and behavior of the VLSI adaptive exponential integrate-and-fire neuron 14

3.2.2 Circuit model analysis and phase portraits 14

3.2.3 Neuron's circuit measurements 19

3.2.4 Thermal noise and mismatch effects 21

3.3 Synapses 22

3.3.1 Spike-Timing Dependent Plasticity learning circuits 22

3.3.2 The spike-based learning algorithm 22

3.3.3 The learning synapse circuit 24

3.3.4 Synapse's circuit measurements 26

3.4 Address Event Representation (AER) 29
3.5 Asynchronous event-based current to frequency converter 30
3.6 Discussion and Conclusions 31

4 Spikebetter: a programmable neural core processor with an asynchronous Static Random Access Memory (SRAM) for enhanced synaptic weights precision
F. Corradi, H. Mostafa, M. Osswald, S. Moradi, and G. Indiveri
34
4.1 Chip architecture 35
4.2 Neurons array 36
4.2.1 Neuron membrane potential 36
4.2.2 Single neuron activation function 37
4.3 SRAM architecture and cell considerations 38
4.3.1 SRAM operations 38
4.4 The synapse Digital to Analog Converter block 39
4.5 Synapse's learning circuit array 42
4.5.1 Long Term Potentiation and Long Term Depression probabilities measurements 43
4.6 Discussion and Conclusions 44

5 A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses
N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri
Frontiers in Neuroscience, 2015 **47**
5.1 Introduction 48
5.2 Neuromorphic processor architecture 49
5.2.1 Synapse temporal dynamics 51
5.3 The silicon neuron block 52
5.4 The long-term plasticity synapse array 55
5.5 The short-term plasticity synaptic array 58
5.6 The peripheral input/output blocks 59
5.6.1 The virtual synapse array 61
5.6.2 AER Input/Output circuits 61
5.7 On-line learning and classification of visual stimuli in a neuromorphic feed-forward VLSI network 65
5.8 Discussion and Conclusions 67

6 Computing with attractor networks on neuromorphic VLSI systems 71
6.1 Attractor Networks as primitives of neural computation 71
6.1.1 The Energy function 72
6.1.2 Effective Transfer Function for large recurrent networks 73
6.1.3 Spike based simulation of an attractor network 75
6.2 Attractor Networks in neuromorphic hardware 79
6.3 Learning associative memories in neuromorphic hardware 80
6.4 Information storage capacity of physical recurrent networks 84
6.5 Discussion and Conclusions 85

7 Decision making and perceptual bistability in spike-based neuromorphic VLSI systems
F. Corradi, H. You, M. Giulioni, and G. Indiveri
International Symposium on Circuits and Systems, ISCAS, 2015 **87**
7.1 Recurrent attractor networks as a model for decision making and perceptual bistability 88
7.1.1 A mean rate model 88
7.1.2 Network architecture 89
7.1.3 Neuromorphic setup 90
7.1.4 The chip architecture 90
7.2 Results: Discrimination of distinct stimuli and basins of attraction 91
7.2.1 Dynamics of perceptual bistability 95
7.3 Long term behaviour and unpredictability of state transitions 96
7.3.1 Message encoding 97
7.3.2 Reconstruction of the attractor 97
7.3.3 Estimation of the largest Lyapunov exponents 100
7.3.4 Recurrence quantification analysis 101
7.4 Discussion and Conclusions 102

8 Computing with the Neural Engineering Framework
F. Corradi, C. Eliasmith, and G. Indiveri
International Symposium on Circuits and Systems, ISCAS, 2014 **104**
8.1 Theory 104
8.1.1 Representation 105
8.1.2 Transformation 105
8.1.3 Dynamics 105
8.2 Experiment: Mapping arbitrary mathematical functions and dynamical system to neuromoprhic VLSI systems 106

8.2.1 Synaptic weights calibration . . . 106
8.2.2 Representations of functions with populations of neurons . . . 107
8.2.3 Real-time computations of mathematical functions . . . 108
8.2.4 Working memory as a dynamical system . . . 109
8.2.5 Discussion: Spike-based distributed analog computing in heterogeneous neuromorphic systems . . . 110

9 A neuromorphic event-based neural recording system for smart Brain-Machine-Interfaces
F. Corradi, and G. Indiveri
Biomedical Circuits and Systems, IEEE Transactions on, (submitted), 2015 **112**
9.1 Introduction . . . 112
9.1.1 Motivations . . . 113
9.2 System and circuits . . . 116
9.2.1 Low noise amplifier . . . 116
9.2.2 Band-pass filter circuit with Address Event Representation (AER) output . . . 117
9.2.3 The asynchronous Delta Modulation A/D converter . . . 118
9.2.4 The AER communication scheme . . . 118
9.2.5 The analog filters: peak detector, trough detector and level crossing 119
9.2.6 The reconfigurable neuromorphic processor . . . 119
9.2.7 The on-chip learning rule . . . 120
9.2.8 The neuromorphic classifier's architecture . . . 121
9.3 Results . . . 122
9.3.1 Low noise amplifier measurements . . . 122
9.3.2 The A/D asynchronous delta modulator . . . 123
9.3.3 Pulse Analog to Digital Converter (ADC) characterization . . . 125
9.3.4 Binary classification of neuro-biological recordings . . . 125
9.4 Discussions . . . 128
9.5 Conclusions . . . 130

10 Towards a neuromorphic vestibular system
F. Corradi, D. Zambrano, M. Raglianti, M. Passetti, C. Laschi, and G. Indiveri
Biomedical Circuits and Systems, IEEE Transactions on, 8:(5) 669-680, 2014 **134**
10.1 Introduction . . . 134
10.1.1 Motivations . . . 135
10.2 A Neuroscientific Model of the Vestibular System . . . 137
10.3 Material and methods . . . 143

10.4 The neuromorphic VLSI multi neuron chip 146
10.5 Results: information processing in the neuromorphic system 148
10.6 Discussion and Conclusions 151

11 Discussion **159**
11.1 Artificial intelligent systems 159
11.2 Small-scale reconfigurable neuromorphic hardware 162
11.3 Benefits and drawbacks of asynchronous logic circuits 162
11.4 Summary and Relevance 163
11.5 Outlook and Conclusions 164

Abstract

The integration of brain-like capabilities in electronic architectures requires understanding of the organizational principles of neurons and synapses in neuro-biological systems. Nervous systems process information in a profoundly different way than standard computer technologies. They exploit billions of slow, inhomogeneous and limited-precision processing units (neurons and synapses) and yet they outperform traditional von Neumann-based general-purpose computers in various problems such as speech-processing, language production, motor control, and multi sensory integration.

Neuromorphic engineering is one of the promising alternative approaches to traditional von Neumann computing architectures. The field takes a different approach than trying to increase the raw computational speed of conventional algorithmic computing systems; it takes inspiration from the efficiency and robustness of biological nervous systems and aims at implementing the same principles of information processing in custom Very Large Scale Integration systems. The challenge of the task is in understanding the complexity of neuro-biological systems together with the integration of experimental finding across multiple levels of investigation. The problem involves findings that range from intelligent behavior to biochemical processes, spanning more than six orders of magnitude at spatial and temporal scale (from nanometers to millimeters and from microseconds to seconds).

Some hints on how to tackle the problem of how neural systems give rise to behavior and computation come from increasing evidence from electro-physiology, neuroanatomy, information theory and computational modeling, which indicate that brains employ algorithms based on hierarchical networks of repeated computing primitives. The search for common circuit motifs in the cortex is deeply appealing to us, neuromorphic engineers, because it supports the idea that we could reduce the apparently insurmountable complexity of cortical wiring into a more regular structure.

Some of the central questions that I investigated are: how do we go from neurons, to primitives of computation, to behaving systems? And in particular, is there any neural circuit template that can be considered as a reusable component in a variety of neural systems? What are the computational primitives that are at the basis of working memory, associative learning and perceptual decision making? How can we implement these primitives in neurally inspired microelectronic circuits?

To answer these questions, I developed (in collaboration with my colleagues) custom mixed-signal analog/digital VLSI hardware architectures. These architectures represent an ideal neural emulation platform as they are compact, low-power electronic implementations of massively parallel distributed spike-based systems. Specifically, I designed and fabricated a variety of biologically inspired circuit blocks that behave in agreement with their neuro-biological counterparts. I addressed the challenge of obtaining distributed and

programmable computation with noisy and heterogeneous analog circuits organized in networks of asynchronous spiking neurons. To this end, I studied the stability of attractor states that emerge in clustered recurrent networks of spiking neurons. I demonstrate how these microelectronic neural circuits can robustly reproduce some of the behaviors observed in neuro-biological networks such as perceptual decision making, working memory, associative learning and binary classification. I established, by means of experiments, that arbitrary mathematical computation in neuromorphic multi-neurons VLSI systems can be achieved using calibration procedures and theory guided approaches.

To further investigate how real neural systems compute, I realized a neuromorphic system capable of recording from neural tissue. This system is conceived as event-based Brain-Machine Interface that exploits asynchronous logic to sense and transmit information collected from neural tissue. The main objective of our design is not a faithful reproduction of action potentials but rather is focused on data compression and fast and efficient processing. This research opens future possibilities in the field of Brain-Machine Interfaces, specifically where there is no need for accurate spike sorting but for a high-dimensional signal that can be compressed and quickly transmitted.

The results of this thesis offer a route towards an optimal design of a new generation of computing technologies that are based on hybrid analog/digital VLSI neuromorphic circuits organized in reusable templates small-scale networks.

Compendio

L'integrazione di capacità cognitive in architetture microelettroniche richiede la compresione dei principi organizzativi e computazionali dei neuroni e delle sinapsi nei sistemi neurobiologici. I sistemi nervosi, come ad esempio il cervello umano, processano l'informazione in maniera profondamente diversa dai computer tradizionali. I sistemi neurobiologici sfruttano miliardi di unità di elaborazione (i neuroni) che processano l'informazione lentamente, in maniera inomogenea e con una precisione molto limitata. Nonostante questo, i sistemi biologici sorpassano di gran lunga i calcolatori tradizionali in diversi ambiti come ad esempio: il riconoscimento vocale, la produzione di linguaggio, il controllo di motori, e l'integrazione di informazione multisensoriale. L'ingegneria neuromorfa si presenta come uno degli approcci alternativi al calcolo e ai calcolatori tradizionali. Questo campo di ricerca studia nuovi metodi per incrementare la potenza di calcolo: si ispira all'efficienza e alla robustezza dei sistemi nervosi con l'intenzione di realizzare sistemi microelettronici che funzionano grazie agli stessi principi computazionali e organizzativi del cervello. La sfida è ardua, e oltre a richiedere la comprensione della complessità dei sistemi nervosi, è richiesta l'integrazione di evidenze sperimentali attraverso diversi livelli d'investigazione. Il problema richiede sia l'analisi di comportamenti cognitivi, che la comprensione dei processi biochimici che stanno alla base dell'elaborazione dell'informazione nei tessuti nervosi. Il problema abbraccia più di sei ordini di grandezza sia su scala temporale, che su scala spaziale; si estende da nanometri (neurotrasmettitori) a millimetri (assoni e strutture multicellulari) e da microsecondi (rilascio di neurotrasmettitori) a secondi (memoria a breve termine), giorni (memoria a lungo termine) o addirittura anni (memorie di infanzia).

Fortunatamente esistono una serie di indizi che potrebbero aiutare a spiegare come i sistemi nervosi danno vita a comportamenti intelligenti. Questi indizi arrivano da una serie di evidenze in diversi campi: dall'elettrofisiologia, alla neuroanatomia, alla teoria dell'informazione e anche da modelli computazionali che indicherebbero che il cervello sfrutta algoritmi basati su reti neurali organizzate in maniera gerarchica, in cui è possibile distinguere strutture primitive di calcolo. La ricerca di primitive computazionali, rappresentate da motivi strutturali nella corteccia celebrale, è profondamente allettante per l'ingegneria neuromorfa, in quanto supporterebbe l'idea che si potrebbe ridurre l'apparente complessità dell'organizzazione celebrale in una struttura più regolare.

Alcune delle questioni centrali che ho investigato durante il lavoro di tesi sono: quale è la relazione tra i neuroni, le primitive computazionali e i sistemi che producono comportamento? E in particolare, esiste una struttura neurale fondamentale che può essere considerata una componente riutilizzabile in diverse reti neurali? Quali sono le primitive computazionali che stanno alla base di processi come la memoria di la-

voro, l'apprendimento associativo e i processi decisionali? È possibile emulare questi comportamenti in sistemi neurali realizzati in microelettronica?

Per rispondere a queste questioni, ho sviluppato (in collaborazione con i miei colleghi) architetture microelettroniche ad integrazione su larghissima scala (VLSI). Queste architetture rappresentano una piattaforma di emulazione ideale in quanto sono compatte, a basso consumo di energia e realizzano sistemi computazionali distribuiti, massivamente paralleli, che comunicano per mezzo di eventi. Nello specifico, ho progettato e testato svariati circuiti neurali che si comportano in accordo con i rispettivi omologhi neurobiologici. Ho affrontato la sfida di ottenere computazione distribuita e programmabile sulla base di circuiti analogici, rumorosi ed eterogenei, organizzati in reti di neuroni basate su eventi. A tal fine, ho studiato la stabilità di stati ad attrattore che emergono in reti di neuroni organizzate in gruppi ricorsivamente connessi. Ho dimostrato come questi sistemi neurali mircoelettronici possano riprodurre in maniera robusta alcune delle dinamiche osservate in reti neurobiologiche, come ad esempio nei processi di decisione percettiva, nei processi di memoria di lavoro, di memoria associativa, come anche nella discriminazione binaria. Ho dimostrato, mediante esperimenti e guidato dalla teoria, che è possibile effettuare calcolo matematico arbitrario per mezzo di reti di neuroni basate ad eventi su chip neuromorfi.

Per indagare ulteriormente come i sistemi neurali processano l'informazione, ho sviluppato un sistema microelettronico in grado di registrare segnali nervosi. Questo sistema è stato concepito come interfaccia tra cervello e macchina. Il sistema sfrutta le potenzialità della microelettronica analogica assieme ai principi neuromorfi per percepire ed amplificare l'informazione collezionata dal tessuto nervoso, codificarla in eventi e trasmetterla a processori neuromorfi. L'obiettivo principale non è la riproduzione fedele dei potenziali d'azione, ma piuttosto è focalizzato sulla compressione dei dati e sull'elaborazione efficiente ed in tempo reale. Questa ricerca apre nuove prospettive nel campo delle interfacce cervello/macchina, nello specifico dove non ci sia la necessità di separare i segnali proveninenti da tanti neuroni, ma dove vi sia l'esigenza di comprimere il segnale neurale e trasmetterlo velocemente.

I risultati di questa tesi rappresentano un passo in avanti verso la realizzazione di una nuova generazione di tecnologie computazionali che si basano su microelettronica ibrida analogico/digitale, che prendono ispirazione dal funzionamento e dall'organizzazione dei sistemi neurali.

Disclaimer

I hereby declare that the work in this thesis is that of the candidate alone, except where indicated in the text and as described below. Chapter 3 contains an extended version of the paper [Mostafa et al., 2014]. Chapter 5 is a modified version of the paper [Qiao et al., 2015]. Chapter 6 contains results published in [Qiao et al., 2015], and in [Indiveri and Liu, 2015]. Chapter 7 extends the work done in the conference paper [Corradi et al., 2015]. Chapter 8 contains the conference paper [Corradi et al., 2014b]. Chapter 9 represents a more detailed version of the work [Corradi et al., 2014a] and it is being extended in a journal publication, currently under review [Corradi and Indiveri, 2015]. Finally, Chapter 10 contains work published in [Corradi et al., 2014c] and [Passetti et al., 2013].

The use of "we" in the thesis refers to the aforementioned people in the relevant sections.

Publications arising from this thesis

The work of this thesis or part of it has been published on journals and conference proceedings as listed below. These publications are also mentioned in the text where relevant.

- F. Corradi, D. Zambrano, M. Raglianti, G. Passetti, C. Laschi, and G. Indiveri, **Towards a Neuromorphic Vestibular System**, *Biomedical Circuits and Systems, IEEE Transactions on, Vol:8, 669–680 Oct. 2014*
- N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, **A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses**, *Frontiers in Neuroscience, Vol:9, Apr. 2015.*
- F. Corradi and G. Indiveri, **A Neuromorphic Event-based Neural Recording System for Smart Brain-Machine-Interfaces**, *Biomedical Circuits and Systems, IEEE Transactions on, (submitted) 2015*
- F. Corradi, C. Eliasmith, and G. Indiveri, **Mapping Arbitrary Mathematical Functions and Dynamical Systems to Neuromorphic VLSI Circuits for Spike-Based Neural Computation**, *IEEE International Symposium on Circuits and Systems (ISCAS), 2014, Melbourne, AU-VIC.* ***(Honorary Mention Award)***
- F. Corradi, D. Bontrager, and G. Indiveri, **Toward Neuromorphic Intelligent Brain-machine Interfaces: an Event-based Neural Recording and Pro-**

cessing System, *Biomedical Circuits and Systems Conference (BioCAS), IEEE, 1-4, 2014, Lausanne, CH.*

- F. Corradi, H. You, M. Giulioni, G. Indiveri, **Decision Making and Perceptual Bistability in Spike-Based Neuromorphic VLSI Systems**, *IEEE International Symposium on Circuits and Systems (ISCAS), 2015, Lisbon, PT.*

- G. Passetti, F. Corradi, M. Raglianti, D. Zambrano, C. Laschi, and G. Indiveri, **Implementation of a Neuromorphic Vestibular Sensor with Analog VLSI Neurons**, *Biomedical Circuits and Systems Conference (BioCAS), IEEE, 1-4, 2013, Rotterdam, NL.*

- H. Mostafa, F. Corradi, M. Osswald, and G. Indiveri, **Automated synthesis of asynchronous event-based interfaces for neuromorphic systems**, *Circuit Theory and Design (ECCTD), European Conference on, IEEE, 1-4, 2013, Dresden, DE.*

- H. Mostafa, F. Corradi, F. Stefanini, and G. Indiveri, **A Hybrid Analog/Digital Spike-Timing Dependent Plasticity Learning Circuit for Neuromorphic VLSI Multi-Neuron Architectures**, *IEEE International Symposium on Circuits and Systems (ISCAS), 2014, Melbourne, AU-VIC.*

- E. Donati, F. Corradi, C. Stefanini, G. Indiveri, **A spiking implementation of the lamprey's Central Pattern Generator in neuromorphic VLSI**, *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE, Lausanne, CH.*

CHAPTER 1

Introduction

1.1 Background

The integration of brain-like capabilities in human made artifacts has been a long-standing effort for quite a while. Since the beginning of the last century scientists have been studying the organizational principles of neurons in neuro-biological systems. The foundations of mathematical models of the electro-chemical synapses, which led Hodgkin and Huxley to be awarded of the Nobel prize in 1963, identifies the action potential (i.e. spike) and its effect on synaptic contacts between neurons as the mean of communications between neural cells. Since then, models based on spiking activity of nervous cells and brain structures have been proposed. Anatomical and functional data from large numbers of cells have revolutionized the understanding of how neural systems integrate a large variety of functionally specialized groups of neurons over many different brain regions. All these complex interactions between brain regions give rise to relevant and adaptive behaviors; how exactly this happens it is still one of the biggest puzzles that humanity is trying to solve.

The interest in understanding how information is processed in neural tissue is not only relevant for understanding how humans, and more generally animals, produce their behavior, but is also crucial in building machines endowed with intelligent capabilities. Many aspects of information processing in neuro-biological systems appear appealing: they are capable of elaborating information in parallel, on demand, with an astonishing energy efficiency and in a fault-tolerant way. These properties have attracted the attention of mathematicians, physicists, biologists, neuroscientists, engineers and computer scientists

and has led to the formation of a new discipline, *computational neuroscience*. This discipline studies the information processing properties of the elementary components of the nervous system, and emphasize a detailed biological description of the physiological, chemical and dynamical properties of neurons and synapses. Modern neuroscientific models connect the microscopic properties, accessible by molecular and cellular techniques, with the system level knowledge that is accessible by the study of behavior.

In the late 1980, Carver Mead together with John Hopfield and Richard Feynman introduced a new concept, which is now a field of research, called *neuromorphic engineering* [Mead, 1990]. They assembled three fields: physics of computation, engineering and neural network. Since then the study of efficient modelling technologies is a key research area for computational neuroscience. The field of neuromorphic engineering describes the use of analog, and more recently also digital, Very Large Scale Integration systems to mimic the behaviour and the computational primitives of neuro-biological systems. To them the use of analog subthreshold transistors operating region of Complementary Metal–Oxide–Semiconductor electronics to realize neurons and synapses dynamics seemed a promising approach to circumvent software limitations. These circuits can be used in parallel, integrated in large scale, resulting in efficient devices in terms of power. A crucial aspect of these devices is the real-time nature of information processing that is implicit in the assumption that every component of the system is active on demand, and that time models itself, as information is present at the time of occurrence of an interesting event (i.e. the transmission of an action potential).

An international community of neuromorphic engineers started to emerge, and standard protocols appeared. One of the most successful protocol describes the use of Address Event Representation, this is a complete asynchronous event-driven communication code in which parallel elements communicate their address to connected units as elementary information exchange (much like the spike). This principle has first been used to implement sensory devices as silicon retinas in which pixels asynchronously report local changes in light intensity by emitting an Address-Event, this signals the location of the cell in the retinal array. Within this approach there is no concept of frame, and events are processed in real-time as they occur. Redundant information that is contained in frame-based systems is no longer sensed, resulting in a natural compression of data as well as in a low-latency sensor.

It soon became evident that, along with the emulation of sensory systems, neuromorphic engineers should emulate elements of information processing downstream from the sensory stage, with the ultimate goal of implementing intelligent functions. To advance in this direction, neuromorphic engineers started building silicon arrays of neurons interconnected by learning synapses. Some fundamental neural circuits, inspired by the

brain's columnar cortex architecture, have been been developed. A recent work shows how by using subnetworks of soft-winner-take-all it is possible to compose arbitrarily complex finite-state machines [Neftci et al., 2013], supporting the idea of the development of generic neural circuits, used as basic computational building blocks for complex information processing systems. Another instance of a reusable component, inspired by the nature of working memory in mammals, is the concept of an attractor network. This is a recurrent neural network in which point attractors are stable configurations of the network dynamics. The basin of attraction naturally implements an associative memory, that has peculiar properties of error correction and pattern completion. The learning of such associative memories has been implemented over stimulus-specific synaptic changes in a neural core processor. This is the first example of unsupervised learning of associative memory in a neuromorphic device [Giulioni et al., 2015].

The research work of this thesis is based on this context, and aims to develop, control and configure novel neuromorphic circuits and systems. I will present different circuits and systems that are part of two distinct neuromorphic devices that have been developed. These circuits represent a step towards endowing VLSI neuromorphic systems with autonomous and robust learning capabilities. The main novel features of these circuits are lower power consumption and better programmability with respect to their precursors. I will also focus on the high degree of freedom of these chips that has led to the applications of these devices in modeling different basic computational blocks of neuro-biological systems. In the second part of the thesis I will focus on how neuromorphic circuits could have an impact in future prosthetic devices and Brain-Machine Interfaces. To this end, we started developing a new generation of neural recording systems that are comprised of a set of circuits to record, amplify, filter and encode neural data in digital asynchronous event-based AER signals. By combining our neural recording systems with neuromorphic architectures we showed how it is possible to process the information from neuro-biological signals with spiking neural network architectures. We demonstrated that it is possible to realize low-power neuromorphic Brain-Machine Interfaces endowed with neural processing abilities, for solving, for e.g., binary classification of spike-patterns.

CHAPTER 2

Neural information processing in physical systems

In recent history many research laboratories as well as private companies have started projects in which the aim is to build a new type of computer that is comparable in form and function to the mammalian brain. This line of research represents a quest for new types of computing technologies as alternatives to von Neumann architecture. This research foresaw unavoidable problems for VLSI technology and a fundamental architectural limitation of the standard von Neumann architecture. In fact as VLSI technology shrinks, noise and faults increase as quantum effects in silicon devices at nano-scale become prominent. In addition all the von Neumann architectures suffer from the bottleneck problem: this is due to the fact that all instructions and all data must pass through the same shared common bus to get in or out of the processing unit. This bottleneck is not only a physical limitation, but has also served as an algorithmic limitation in controlling the way in which we program and compute with these type of architectures [Backus, 1978].

2.1 Cortical computation and canonical macro circuits

Cortex is present in all mammals and it is a valid candidate for the neural structure that is responsible for human intelligence. It plays a central role in different functions as memory, associations, attention, perceptual awareness, language, motor control, and consciousness. In 1955, [Mountcastle et al., 1955] recorded cells in the cat sensory cortex. He observed that neural cells were separated into domains responsible of different sensory modalities. In his work he hypothesized the existence of elementary entities

of organization in the sensory cortex that are made up of vertical groups of neurons stretching through all the cellular layers. Only about a decade later, [Hubel and Wiesel, 1962] introduced the concept of signal processing within a column. This concept was based on the neurophysiology of cat visual cortex and led them to be awarded the Nobel prize in 1981. Today understanding of the cortex is based on the idea that it is composed of discrete, modular columns of cells, each of which is similar in the way it is wired up, and every column is responsible for a particular type of computation [Douglas et al., 1989]. If one has to look at a pool of cortical neural cells from the top would notice that their axons project to neighbouring cells as "petals of a flower" [Douglas and Martin, 2004]. The current view is that these clusters of axons convey activity of pyramidal cells to collectively participate in a selection network. This notion of cortical macro circuits is deeply appealing to neuroscientists and engineers as it would suggest that we could reduce the apparently insurmountable complexity of cortical wiring into a regular array of similar columnar arrangements. This idea is pursued by many neuroscientists that aim at understanding or reproducing computational and signal processing abilities of the cortex [George and Hawkins, 2009; Kaschube et al., 2010; Perin et al., 2011]. There is however a controversy in the columnar organization of the cortex, as some species, such as rodents, do not seem to have a clearly arranged orientational column described by Hubel and Wiesel. Another aspect that may cast some doubt on the canonical circuits idea is the anatomical difference between brain regions and between species. These differences pose the question of whether the columnar modules are necessary for cortical functions [da Costa and Martin, 2010; Horton and Adams, 2005]. Nevertheless today's understanding of the cortex is based on two main concepts. First, cells that share receptive fields are often clustered together. Second, cellular connections within and between different regions are often regular. These circuits might not be strictly organized into well-defined columns, as in rodent, but in a more heterogeneous fashion. The search for common patterns in the organizational principles of the structure of the cortical area between species evidence interesting perspective. In this context the concept of common reusable computational blocks, i.e. macro column, seems to suggest a hierarchical computing platform that profoundly differs from von Neumann computing architectures. In addition, recent work in the direction of evolutionary algorithms [Ellefsen et al., 2015] suggests that neural modularity would help organisms to learn new skills without forgetting old skills. This result was produced by letting different neural networks evolve that would differ in structure; more modular (more clustered interconnections) or less modular (more homogeneously distributed connections); and measuring at which point catastrophic forgetting would happen. It turns out that networks organized in a modular manner, would tend to reduce catastrophic forgetting because skills can be

learned in different modules, supporting the idea of modular circuits used as primitives of computations. The challenge of an alternative computational architecture has been addressed at different scales, and with different technological implementations that ranges from completely digital unconventional programmable architectures, to analog neuromorphic wafer-scale systems. The common line of research in all these architectures is that they are inspired by the spike-based information processing abilities of neural systems, and by their organizational principles, as in the cortex. In the next two sections I present a survey of systems, proposed in laboratories and industries around the globe, whose target is the simulation (or emulation) of spiking neural networks.

2.2 Large-scale spike-based computer models

Spiking large scale neural network simulations on general platform computers have been adopted in the recent past. One of the recent large-scale models simulated a subset of neurons of the mammalian thalamocortical system [Izhikevich and Edelman, 2008]. In this model they used one million neurons and 22 types of different neuronal compartmental models with 500 million synapses. To this end they used 60 general purpose computer at 3GHz processor with 1.5 GB of Random Access Memory (RAM) each. The program was written in the C language with Message Passing Interface (MPI) and was running on a Beowulf cluster. Their model exhibited phenomena such as spontaneous activity and rhythms of spiking activity, that are compatible to what has been observed by means of electro-physiological recordings.

One of the most highly publicized large-scale simulation platforms that is being developed is within the Human Brain Project, which started in 2005 as the Blue Brain Project [Markram, 2006]. Recently the project was awarded one billion euros of funding from the European Union. One of the goals is the creation of a platform that integrates neuroscientific data such as gene expressions, cell types, regional connectivity, etc., in a single massive database. Together with the development of a general atlas of the brain, another initial goal of the project is to build a working simulation of the entire human brain. The model used is focused on simulating cortical columns, described as cylindrical groups of $100,000$ neurons that make up the cerebral cortex of mammals. The ambitious project aims at creating a large-scale Brain Simulation Platform. The platform should make it possible to build and simulate unifying brain models that would integrate all the available data, generating emergent structures and behaviors that could not be predicted from smaller data sets. The major part of the data included in the database are gathered from rodent slice experiments and connectivity is determined by the statistical properties of observed connectivity across slices [Hill et al., 2012]. The largest simulation of this model have included about a million neurons, each neuron and

synapse have been simulated with a great level of detail, taking into consideration ion channel composition, spatial morphology and detailed physiological data.

In the United States of America another large-scale model project has been developed, and now reached its end. This project was sponsored by DARPA and was called the Synapse project [Preissl et al., 2012]. The cortical model is based upon the work presented in [Ananthanarayanan and Modha, 2007], this work presents a simulation of hundreds of millions of cortical neurons. In contrast to the European Human Brain Project the neuron model used in the Synapse project is much simpler and only includes neural spikes and spike-timing-dependent plasticity (STDP), and very little or nothing about spatial morphology or genes expression. As a result many neurons can be simulated and today they have reported a model with $53\text{x}10^{10}$ neurons, and $1.37\text{x}10^{14}$ synapses running 1542x slower than real time. The number of synapses are in the same order of magnitude as in the human brain [M. Wong et al., 2013].

2.3 Large-scale neuromorphic hardware systems

Hardware implementations of large-scale neural models seek to integrate experimental findings across multiple disciplines in order to explain how adaptive and intelligent behavior in neural systems emerges. To this end a high computational power budget is required. The simulation of a human-scale cortex model ($2\text{x}10^{10}$ neurons i.e. the Human Brain Project's initial goal), would require $8\text{x}10^{16} flops$ [1] and as much power as $0.5\text{x}10^{10} Watt$ [2], ie $5GW$ (nowadays, this much power can be provided by the world's largest coal-fired power station). Therefore several groups around the globe are developing custom large-scale neural hardware emulators that would simulate the largest possible number of neurons and synapses at an affordable power budget. The SpiNNaker project [3]: aims to develop a platform based on 18 mobile processors into a single die [Furber et al., 2014]. These chips includes 18 general-purpose ARM9 [4], with 100kByte local memory for each core (program and data) and 128MByte shared Dynamic Random Access Memory (DRAM) used for storing synaptic weights or other internal variables. These cores provide a programmable platform for implementing arbitrary neuron models and learning algorithms in which computation is based on a 16-bit fixed point arithmetic.

[1] In the simplest model possible, we could consider that the brain is composed of 80 billion neurons, with one thousand input connections in average per neuron. If we store all the $8\text{x}10^{13}$ connection weights in 32 bit floating point and we use a simple IF neuron to run the model we will need $8\text{x}10^{13}$ floating point operations per 'brain cycle'. If we assume that the most fastest neurons could fire every ms, we then would need $8\text{x}10^{16}$ flops, i.e. 80 Petaflops

[2] We used the IBM Blue Gene measure from 2010, i.e. 1,684 MFLOPS/watt

[3] Led by S. Furber at The University Of Manchester

[4] Advandec RISC Machine (ARM): a family of RISC-based processor designed and licensed by ARM Holding. ARM9 is a 32-bit ARM mobile processor.

If a simple point-neuron model is used every ARM core could in principle implement 1000 neurons in real time. Another aspect that allows the simulation of a large number of neurons is the particular communication protocol and the architecture. In fact, the SpiNNaker system is Globally Asynchronous Locally Synchronous (GALS) system. This is achieved thanks to the custom-designed digital asynchronous logic that is used to pass messages between distinct ARM9 core processors. Each chip contains a router with six input and output ports that connect each chip to other six adjacent chips. One of the declared goals of the project is to simulate one billion neurons organized in a toroidal network structure. A SpiNNaker board with 18 ARM968 processors have recently been used as a demonstrator with a peak power consumption of 1 W [Painkras et al., 2013]. The NeuroGrid project [5] uses analog sub-threshold circuits as custom implementations of neurons and synapses. One of the main goals is to study some of the principles of operation of the mammalian cortex in an affordable hardware system. The Neurogrid system is composed of sixteen neural core chips each containing 65000 neurons in a silicon area of 12x14mm [Choudhary et al., 2012]. The full system contains 1 million neurons, modeled as quadratic integrate-and-fire neurons. To enable the system to reach large scale (1 million neurons) there are several trade-offs. As for example, neural plasticity in Neurogrid is not implemented. Also the asynchronous routing scheme supports a limited amount of traffic. Recently [Choudhary et al., 2012] have used the system, together with the Neural Engineering Framework (NEF) [Eliasmith and Anderson, 2003] a theoretical framework to translate equations and dynamical systems into neural network dynamics. They achieved reliable computations and working memory dynamics using 4000 neurons with 16M feed-forward and off-chip recurrent connections. The Synapse project [6] aims to overcome the memory bottleneck problem by using custom digital circuits that uses virtualization and share resources to emulate a single neuron. The system architecture, called TrueNorth, is built from a multi neurosynaptic core system. Each neurosynaptic core comprises 256 neurons and 256x256 finite-resolution synapses [7] using a cross-bar array [Arthur et al., 2012]. Together with the fully custom and digital neurosynaptic core, IBM developed a TrueNorth software architecture (called Compass) that can be used to reproduce and simulate in software the behaviour of the neurosynaptic core, their impressive results show simulations of $5.3\text{x}10^{10}$ neurons interconnected by $1.7\text{x}10^{14}$ synapses [M. Wong et al., 2013]. The FACETS/BrainScale project[8] aims to achieve integration by using an analog approach to computation, in which neurons are simulated

[5] Led by Prof. Boahen at Stanford University

[6] IBM Golden Gate Chip

[7] Several versions of neurosynaptic cores have been developed, in some there is a 1-bit precision synapse in other 4-bit precision synapses are used.

[8] Heidelberg University BrainScale project

from 10^3 up to 10^5 times faster than real-time. The approach is based on a full wafer system composed of multiple analog neuromorphic dies communicating using a digital asynchronous protocol [Schemmel et al., 2010]. The system includes learning capabilities via Spike–timing–dependent Plasticity learning circuits that involve computation of spike-time difference in analog circuits while the synaptic update is delegated to additional digital external circuits. The large variability of the neurosynaptic core offers limited precision in the mapping of synaptic weights or network dynamics. Classification using a single die chip has been demonstrated in a model of the olfactory system in which weights were programmed after an off-line procedure [Schmuker et al., 2014].

2.4 Reconfigurable hardware for spiking-neural networks

Field Programmable Gate Array (FPGA) digital circuits have had a large impact on the development of custom digital chips by enabling a designer to try custom design on easily reconfigurable hardware. In fact, FPGAs contain programmable logic blocks that are based on large number of gates and local memory, and connections between blocks are configurable. This reconfigurability makes it possible to realize parallel systems such as Spiking Neural Networks (SNN) in hardware [Cassidy and Andreou, 2008; Cheung et al., 2012; Maguire et al., 2007; Wang et al., 2014; Wood et al., 2012]. The common feature of these spiking neural network architectures is that memory is distributed among massively parallel configurations of elementary computational blocks: neurons and synapses. At present, current FPGA systems are capable of simulating up to $4k$ neurons in real-time by exploiting time-multiplexed resources on the board (Virtex 6) as in [Wang et al., 2013], with the ultimate intent of simulating polychronous spiking neural networks to store spatio-temporal spike patterns. Another example is the application of FPGAs in modeling real-time Spike–timing–dependent Plasticity (STDP) [Belhadj et al., 2008]. In this study they used a commercial Spartan 3 FPGA and they achieved 625 synaptic connections in a network composed of 25 neurons. Another alternative approach to reconfigurable hardware for neural network is the use of custom Field Programmable Analog Array (FPAA)s. This approach is mainly pursued by a laboratory situated in Virginia [9]. Their custom VLSI systems are based on low-power analog signal processing blocks. They successfully implemented a linear and non-linear classifier based on a vector-matrix multiplication and a winner-take-all network [Hasler et al., 1998]. The custom chip includes basic programmable amplifiers, filters, adaptive filters, multipliers and gain controlling circuits. Interestingly, the same approach has also been applied in the field of neuroprosthesis. Specifically in a work by [Hogri et al., 2015], they implemented

[9] Cooperative Analog and Digital Signal Processing : Georgia Institute of Technology

a model of the cerebellar cortex in a FPAA [Bamford et al., 2012]. In this work they successfully demonstrated the ability of the hybrid system to learn an association in a classical conditioning experiment that was carried out in anesthetized rats. Many laboratories are building programmable custom mixed-signal analog/digital VLSI circuits to mimic neuro-biological behaviour of neurons and synapses. These custom made chips usually counts about 100 neurons and thousands of synaptic contacts [Badoni et al., 2006; Bamford et al., 2013; Chicca, 1999; Indiveri et al., 2011; Moradi and Indiveri, 2011]. In Rome, at the Istituto Superiore di Sanità[10] a research lab studies neural dynamics, and in particular the research is focused on learning algorithms implemented in a neuromorphic spiking neural network. They successfully demonstrated robust learning of associative memories via an unsupervised learning protocol. The neuromorphic system was composed of 196 neurons with about 128K plastic bistable synapses [Giulioni et al., 2015] and a neuromorphic vision sensor [Lichtsteiner et al., 2008]. Their results were achieved using the framework of *mean-field* theory [Fusi and Mattia, 1999]. This work represents a crucial step in the realization of basic reusable network components that can be combined as fundamental building blocks for implementing more complex cognitive functions, such as associative memories, attention selection, decision-making and choice behavior [Amit and Brunel, 1997; Rolls and Deco, 2002; Wang, 1999].

2.5 Discussion: the right question for the appropriate scale of investigation

It is often assumed as in the case of some of the large-scale projects, that the bottom-up approach, in which as many details as possible are incorporated in a huge simulation, would allow researchers "to study the steps involved in the emergence of intelligent behavior" [Markram, 2006]. One criticism to this approach is the fact that it relies on the notion of emergence to produce behavior and it is not clear how simulating neurons with similar statistical properties in form and structure to the cortex will allow for interesting behavior to emerge. In contrast, there are different groups following a top-down line of research in which they first identify behavioral functions of a certain brain area and sequentially propose models of how neural circuits can accomplish that particular function [Deco and Rolls, 2002; Deco et al., 2004; Del Giudice et al., 2003; Eliasmith et al., 2012; Maass, 2000; Wang, 2012]. Within this approach the use of models to test hypotheses about the function of different brain regions is essential. In principle if behavior is not contemplated within the modeling hypothesis it would be mysterious if it would emerge from a collective statistical mixture of components. Without a clear link

[10] Complex System Laboratory Group, led by Prof. P. Del Giudice

on causes and effects we could miss the fundamental purpose of all neuroscience. Another important aspect to consider is the level of details at which we are posing a question. There is always a trade-off between the amount of details, the computational resources and the questions that need to be answered. All the projects presented are competing to find the right balance between model complexity and efficiency; some use the bottom-up approach and others the top-down. The level of investigation at which the work of this thesis focuses, is on the small scale networks of spiking neurons (typically hundreds) with thousands of synapses with limited but distributed precision on the synaptic weights. With such small systems we investigated many neural circuits that are used to model different regions of the nervous system. For example, we studied the mechanisms that link recurrent network dynamics to behavioral experiments involving perceptual decision making (Section 7). Another example is the development of a neuromorphic model of the vestibular system by using a network of spiking neurons distributed over two neuromorphic devices together with a commercial Inertial Measurement Unit (IMU) sensor (Section: 10).

CHAPTER 3

Electronic neurons and synapses

Neuromorphic engineering aims at emulating the organizing principles of nervous systems in VLSI hardware by exploiting physical properties of Complementary Metal–Oxide–Semiconductor (CMOS) semiconductor devices [Mead, 1989]. The architecture of these hardware implementations directly relate to neurobiological systems in different aspects. Similarly to the nervous system that carries out computation in a robust and reliable manner using many slow and unreliable processing elements, we are developing a new generation of computing technologies exploiting reconfigurable custom mixed signal analog/digital systems that implement several models of cortical-like computation. In particular we empathize distributed, event-based, massively parallel and collective mechanisms that exploit adaptation and self-organizational principles for learning. One of the main features of the neuromorphic systems that we develop is that they can be optimized for real-world applications such as robotics, prosthesis or biomedical applications. In these fields of operation real-time interaction with sensory systems is a requirement, as well as optimal performance in power-consumption and a high level of integration. In this chapter we describe some of the circuits and methodology developed in order to map the biochemical processes of neurons and synapses with the physics of transistors.

3.1 Motivations: Mimicking neurobiology with analog neuromorphic VLSI circuits

There exist many analogies among biological neural systems and mixed signal analog/digital VLSI neuromorphic systems. These analogies are based on the concepts of conser-

vation of charge, gain control, compression of information, integration, threshold and non-linearity. These concepts are applied at different scales: from the device physics to circuit architectures. In fact, in the neuronal ion channels as well as in transistors there is a diffusion mechanism that modulates the channel. The drift and diffusion equations, that are used to describe the flow of charge in a channel, have the same exponential distribution of particles in both cases. We refer to ions in biology and to electron-hole pairs in silicon. If we look at the actual spatial scale they also almost match as synapse channels and transistors are both in the sub-micrometer scale. In both cases the behaviour of the circuit is not determined by an algorithm but by the structure of the system and its dynamics. Performing a portion of the computation in the analog domain is advantageous because analog circuits readily perform biology-like computing, as the exponential flow of charge in a channel relates to the sub-threshold current of a transistor. The same computation can be demanding if carried out with digital circuitry [Mead, 1989] [Frantz, 2000]. In addition the study of neural systems models can potentially be beneficial for improving current technologies. To have a grasp on that we must consider that biology appears to have solved the problem of using massively parallel, noisy, globally asynchronous and loosely coupled components to carry out robust, multi-tasking and energy-efficient computing. Additionally CMOS VLSI technology has been constantly improving at a paste that have been exponential for almost 50 years [Mack, 2011]. But as CMOS keeps improving and transistor gate length constantly drops, undesirable effects start to emerge. These effects can be seen as if transistors start to become more neuron-like in the sense that they are less reliable and more faulty. We now have the ability to place millions of simple computing processors in a single die. We are aware of the astonishing power dissipation of the brain, approximatively $20W$, and of its capability of processing an equivalent of something like $8\text{x}10^{16}$ floating point operations per second (see Section 2.3). This tell us that the brain is therefore consuming about $2.5\text{x}10^{-16}J$ per operation. Today custom digital systems are many orders of magnitude away from that, as for example the world fastest super computer the 'Thiane-2'[1] that consumes about 800KW[2] and can achieve at best $33.86\text{x}10^{12}$ flops. All this results in a consumption of $2\text{x}10^{8}J$ per floating point operation. If one considers that information in neural system is transferred by means of spikes, and therefore the comparison is not fair, even if we consider the IBM [Merolla et al., 2011] neurosynaptic digital custom system, which is the best achievement in term of efficiency in power per spike, it results in a consumption of $45x10^{-12}J$ per synaptic event. This is still far away from the efficiency observed in a biological neural system. Having all this in mind, our quest is focused

[1]Thiane-2 is a cluster of 16.000 computers. They use Xeon Intel processors that have been fabricated in 22nm CMOS technology.

[2]assuming that each CPU consumes in average 50W in an optimistic measure

on finding elementary neuro-inspired computing elements for constructing efficient and intelligent systems. In particular we focus on ultra-low-power implementations of neuron and synaptic circuits, that we choose by design to be implemented in analog low-power subthreshold circuits. With the term subthreshold we refer to the region of operation of transistors in which is exploited a conduction process without fully turning on or off the transistor. In this operational regime currents are in the order of pico Amperes[3] resulting in an ultra-low-power design.

3.2 Neurons

3.2.1 Model and behavior of the VLSI adaptive exponential integrate-and-fire neuron

Compact VLSI hardware implementations of spiking neurons with biophysically realistic dynamics attempt to achieve a good compromise between power efficiency and compactness on one hand while trying to deliver a rich dynamical spiking behavior on the other. In this chapter we analyze the adaptive exponential integrate-and-fire neuron circuit as a dynamical system, with particular interest in determining the crucial parameters and their respective relevant value ranges for obtaining a variety of biologically realistic dynamics. The current-mode neuron circuit is implemented in the subthreshold region of operation of transistor and therefore it is modelled by a highly non-linear system of differential equations, as the current through the channel exponentially depends on the gate voltage.

3.2.2 Circuit model analysis and phase portraits

The adaptive exponential integrate-and-fire neuron circuit we describe here, and visible in Fig. 3.1, is originally presented in [Qiao et al., 2015] and it is the successor of the circuit first introduced in [Livi and Indiveri, 2009]. It is a phenomenological silicon neuron with bio-physically realistic dynamics, such as spike-frequency adaptation, refractory period mechanism and adjustable spiking threshold mechanism. The circuit comprises five blocks:

- an inputDifferential Pair Integrator that emulates the effect of the NMDA (N-methyl-D-aspartate) postsynaptic-receptor voltage gating.
- A leak term with two programmable leakage parameters (if_tau1, and if_tau2).
- an adaptation block that implements spike-frequency adaptation with adjustable gain (if_ahthr) and time constant (if_ahtau).

[3] if we use 180nm technology

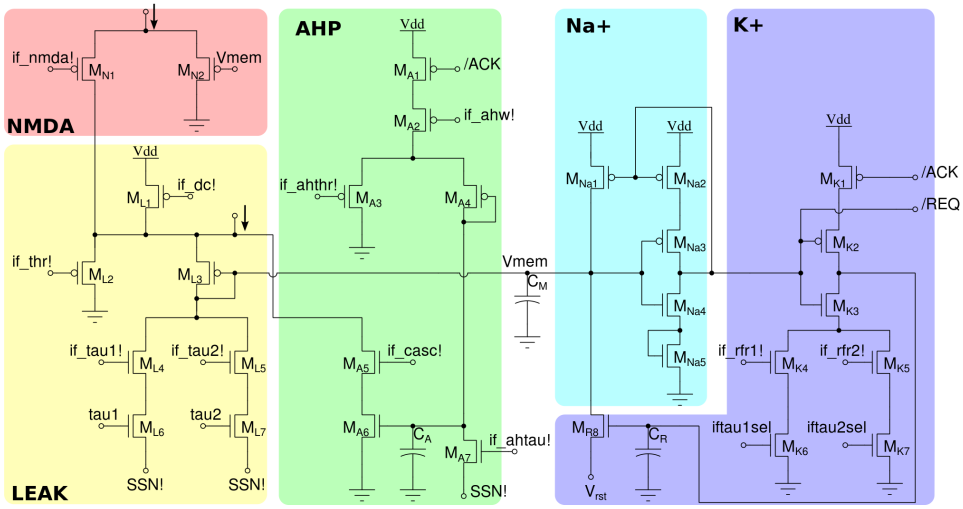


Figure 3.1: Silicon neuron schematic. The NMDA block implements a voltage gating mechanism; the LEAK block models the neuron's leak conductance; the spike-frequency adaptation block AHP models the current effect of the post hyper-polarization; the positive-feedback block Na^+ models the effect of the Sodium activation and inactivation channels; the reset block K^+ models the Potassium conductance functionality.

- an $Na+$ positive feedback circuit which models the effect of Sodium activation and inactivation channels for spike generation mechanism
- a $K+$ with an adjustable refractory period mechanism (if_rfr).

The behavior of the silicon neuron is determined by the leaky conductance that produces exponential dynamics in response to constant input current. The input current is integrated in the capacitor C_m, which represents the neuron's membrane capacitance; this integration leads to the increase of the membrane voltage. Meanwhile, the positive feedback circuit, which mimics Sodium ($Na+$) activation and inactivation channels in real neurons, provides larger current to the membrane capacitor for higher membrane voltage, and further accelerates the increasing of the membrane voltage until a threshold is reached. When the threshold of the first inverter is reached, a digital event (i.e. a spike) is produced. The spike reset block with refractory period functionality forces the membrane voltage to stay at the reset potential for a finite amount of time. The last inverter performs the generation of a digital spike.

Phase portraits analysis

We analyze the system of differential equations governing the subthreshold neuron dynamics with the goal of determining the neuron's dynamical properties. We apply

Dynamical Systems Theory for this task. We use the straight-forward method of drawing the nullclines of the variables into phase space and we draw sample trajectories for varying parameters. In this way, we can determine the crucial parameters and their relevant value ranges for producing biologically realistic neural dynamics. The phase portrait contains information about location and stability of equilibria (resting states), limit cycles (phasic spiking) and separatrices (thresholds).

Differential equations and approximations

$$\frac{dI_m}{dt}(t) = \frac{I_{pos}(t) - I_{mem}(t)(1 + \frac{I_{ahp}}{I_\tau})}{\tau_m(1 + \frac{I_{th}}{I_{mem}(t)+I_0})} \tag{3.1}$$

$$= \frac{(I_{mem}(t) + I_0)(I_{mem}(t) + I_{th})}{I_{tau}(I_{mem}(t) + I_0 + I_{th})\tau_m} \left(\frac{I_{again}}{1 + e^{\frac{I_{ath} - I_{mem}}{I_{anorm}}}} + \frac{I_{in}(t)}{1 + \frac{I_{mem}}{I_{th}}} - I_\tau \right) \tag{3.2}$$

$$\frac{dI_{ahp}}{dt}(t) = \frac{I_{posa} - I_{ahp}(t)}{\tau_{ahp}} \tag{3.3}$$

with:

$$I_{pos} = I_{fb} + \frac{I_{th}}{I_\tau}\left(I_{it}(t) - I_\tau - I_{ahp}\right) \tag{3.4}$$

$$I_{fb} = \frac{I_a}{I_\tau}\left(I_{mem}(t) + I_{th}\right) \tag{3.5}$$

$$I_a = \frac{I_{again}}{1 + e^{-\frac{I_{mem}(t) - I_{ath}}{I_{anorm}}}} \tag{3.6}$$

$$I_{posa} = \frac{I_t ha}{I_{\tau a} I_{wa}} \tag{3.7}$$

A nullcline contains all the points in phase space, for which the variable in question is at rest. The nullcline for the system of equations 3.1 for I_{mem} and I_{ahp} are:

$$I_{ahp}(I_{mem}) = \frac{I_{in}(t)}{1 + \frac{I_{mem}}{I_{th}}} + \frac{I_{again}}{1 + e^{\frac{I_{ath} - I_{mem}}{I_{anorm}}}} - I_\tau \tag{3.8}$$

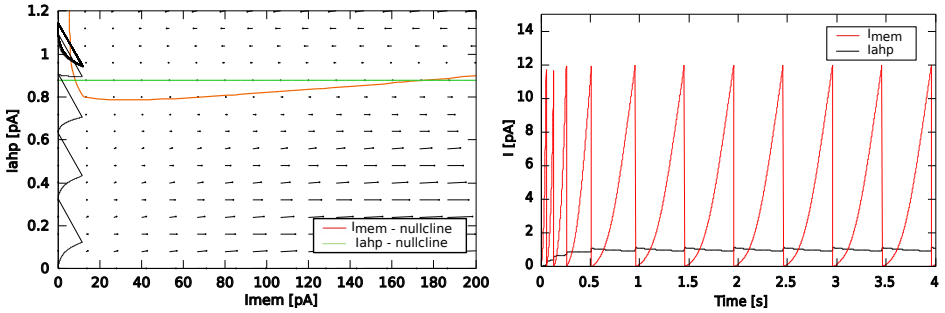
$$I_{ahp}(I_{mem}) = \frac{I_{tha}}{I_{\tau a}} I_{wa} \tag{3.9}$$

The structure of I_{mem} nullcline is composed of three different terms:

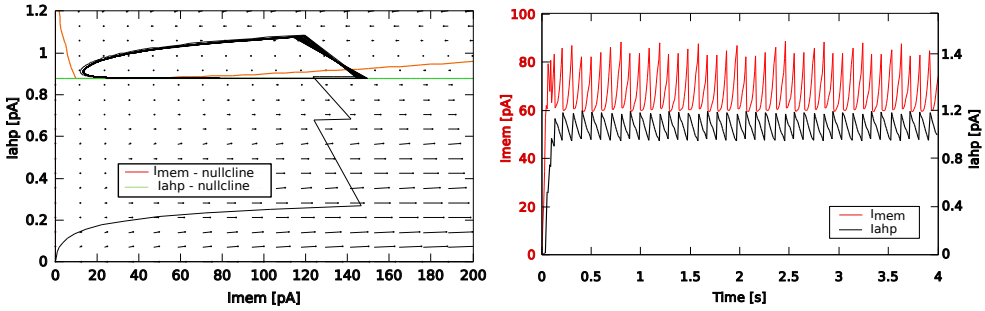
- A rational function: $\frac{1}{1+x}$, with pole at -1 (outside the region of interest) and asymptote $y = 0$. For the relevant interval (positive currents, $x > 0$) this behaves as a negative exponential e^{-x}.

- The Boltzmann-function $\frac{1}{1+e^{a-x}}$ which causes a steep rise of the nullcline for larger I_{mem} values.
- A constant leak term $-I_{tau}$

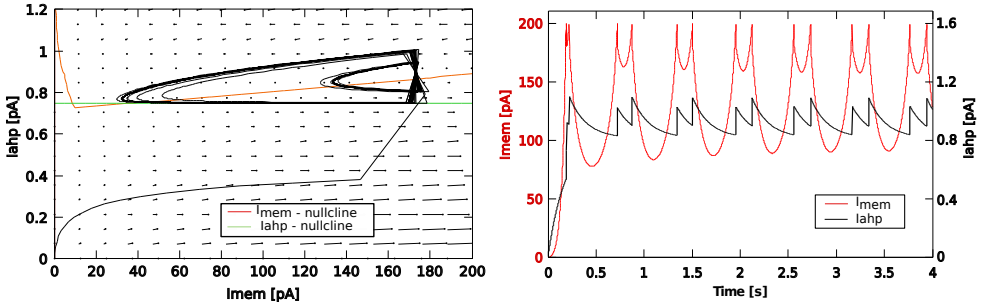
These terms act on different regions of the phase space. The third term has a global influence while the first two terms act on local regimes. The vector fields for these equation is shown in figure 3.2. By varying the parameters as in table 3.2.2 it is possible to shift the nullcline in 3.2 and obtain three different situations. By doing so, we obtain the phase plots visible in 3.2. In 3.2a the two nullclines touch in two distinct points. By injecting a constant current of $6pA$ in the neuron's soma, I_{ahp} starts to increase until the point at which it oscillates around a mean value. This is caused by the reset mechanism of the neuron. The initial low value of I_{ahp} current allows the neuron to fire at an initial higher frequency. When I_{ahp} reaches about $1pA$, visible in 3.2a (left and right), the system reaches and oscillating equilibrium in which the neuron fires at a constant frequency. This behavior can be classified (according to [Izhikevich, 2006]), as type *class I excitatory* neuron. A different behavioral situation emerges from Fig. 3.2b, also in this case the adaptation current undergoes an initial value increase under excitation. However, in this case the two I_{mem}, I_{ahp} nullclines touch in a single point, and this does not allow the membrane potential of the neuron to be reset at $0V$. In this behavior the neuron cannot asymptotically approach $0Hz$ of mean frequency as there will always be a minimum input current that will make it fire at a rate that is well above $4Hz$. This behavior can be classified as a different neuron type than the previous one, and more specifically as *class II excitatory* neuron. The third possible behavior with this system of equations is the *bursting* neuron. This is shown in Fig. 3.2c. In this behavior the nullclines have two points of contact, very close to each other ($I_{ahp} \approx 0.7pA$, $I_{mem} \approx 10,25pA$). This behavior is obtained by setting the value of the reset current close to the firing threshold of the neuron. This means that, as you can see from Fig. 3.2c left, once every two spikes the membrane potential is reset below the current reset value of $I_{reset} = 180pA$, because I_{ahp} sinks current from the membrane potential after a spike event. Once every two spikes I_{ahp} current is higher than the current at the resting state I_{reset}, causing the neuron to emit two fast spikes followed by a longer period of silence.



(a) Class I neuron



(b) Class II neuron: the minimum firing rate is $4Hz$.



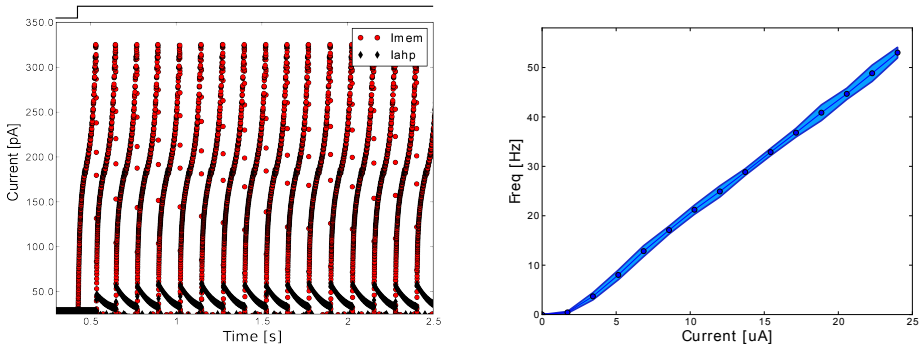
(c) Bursting

Figure 3.2: Neuron behaviour and phase space

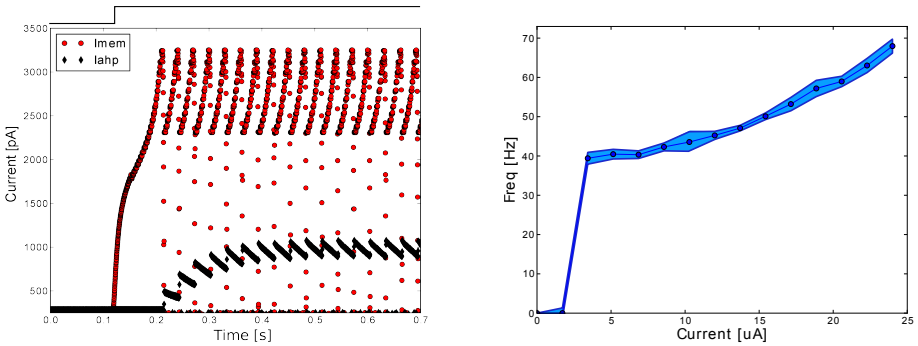
	Current Value [pA]		
Current Name	**Class I**	**Class II**	**Bursting**
I_{in}	6	0.2	0.05
I_{reset}	50	60	180
I_o	0.5	0.5	0.5
dI_{ahp}	1	0.2	0.2
I_{th}	100	89	200
I_{tau}	0.3	0.002	0.008
I_{tha}	70	1.1	1.2
I_{taua}	1.3	0.05	0.1
I_{wa}	0.1	0.04	0.07
I_{again}	1000	1000	1000
I_{ath}	6000	7200	7100
I_{anorm}	1000	1000	1000

3.2.3 Neuron's circuit measurements

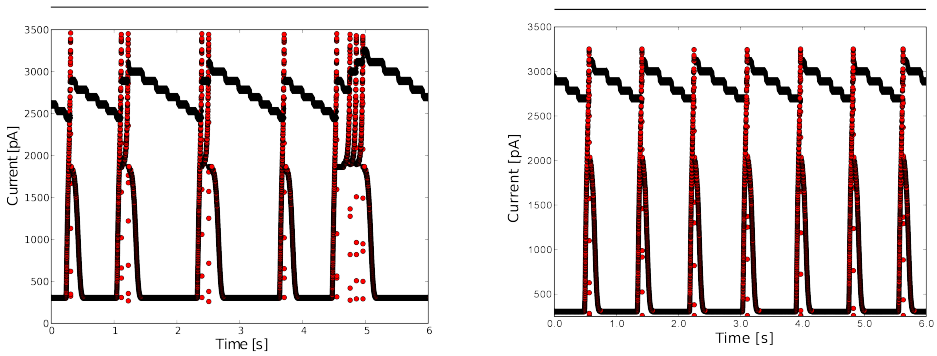
The adaptive and exponential integrate-and-fire circuit described in this chapter has been fabricated in a Complementary Metal–Oxide–Semiconductor $180nm$ technology. Measurements for the different observed dynamical behaviors are shown in figures 3.3. The three different behaviors of the neuron have been observed with slightly different parameters that previously presented in out analysis and visible in table 3.2.2. We allege this difference by our simplified model of equations, in which we do not take into account second-order effects such as as the Miller effect, or transistor sizes. However, the theoretical analysis indicates the ranges of parameter values and it has be used as a tool to identify interesting regions in the vast parameter space. Figure 3.3 shows the result of some measures. The current values have been measured with the use of a current to frequency converter circuit described in Section 3.5. Figure 3.3a evidences the membrane potential and the adaptation current of a neuron of first type, in which the frequency linearly increases with the current input 3.3a. Notice that the first few spikes have a shorter Inter–Spike Interval (ISI) than the latter, this reflects *spike frequency adaptation*. The measure in 3.3b evidence a type II excitatory neuron in which the reset potential is close the spiking threshold. The minimum firing rate of the neuron is about $4Hz$, reflecting the behaviour predicted by our previous analysis in Fig. 3.2b. Bursting neurons have also been observer as in Fig. 3.2c. However an irregular and a more regular behavior have been found. In the next paragraph we explain what are the differences among the two behaviors and their cause.



(a) Spike Frequency Adaptation, the blue shadow is the standard deviation over five different trials.



(b) Class II, the neuron shows a minimum firing rate frequency of $4HZ$, after which it become silent. The blue shadow is the standard deviation over five different trials.



(c) Bursting: irregular and regular.

Figure 3.3: Neuron behavior and phase space

Current Name	Current Value			
	Class I	**Class II**	**Bursting**	**Regular**
RST	48.05 pA	1.61 uA	18.54 nA	18.16 nA
$ATHR$	9.70 nA	9.76 nA	1.26 nA	1.26 nA
RFR	14.06 uA	14.06 uA	5.34 nA	5.34 uA
AHW	1.50 uA	1.50 uA	23.81 nA	23.81 uA
$AHTAU$	1.23 pA	1.23 pA	0.82 pA	0.82 pA
DC	0.312 uA	4.00 uA	7.4 uA	7.4 uA
TAU	32.03 pA	32.03 pA	3.17 uA	3.17 uA
$NMDA$	8.9 nA	8.9 nA	23.9 uA	23.9 uA
$CASC$	23.91 uA	23.90 uA	23.9 uA	23.9 uA
THR	102.50 pA	102.5 pA	1.14 nA	1.14 nA

3.2.4 Thermal noise and mismatch effects

Noise plays a crucial role in the dynamics of the third behavior of the neuron: the bursting. From the phase space plot we note that the two nullclines touch in a single point. The position of the two nullclines non-linearly depends on the parameters of the silicon such as temperature and electrical noise. Note that to obtain the bursting behavior, see Fig. 3.3c, the neuron is firing a couple of spikes interleaved by a precise period of time. With the measurements taken by a fabricated neuron in Complementary Metal–Oxide–Semiconductor $180nm$ technology we obtain similar results. The difference is in the fact that we do not obtain a precise pattern, but rather a more noisy version. The dynamics are governed by the effects of the adaptation time constant that sinks current from the membrane capacitor, it decays linearly and it is incremented upon every spike event. The small difference between the two peaks in the I_{ahp} trace in Fig. 3.3c is amplified by the minimum (at about $100pA$) of the membrane current: the highest the adaptation current the lower the membrane potential will be driven. In advanced computing technologies reliability is starting to represent a serious limiting factor, this is caused by the very small scales ($\sim 20nm$) of higly integrated substrates. In biology and especially in neural systems, variability and stochastic behaviors are present at all levels. From the stochastic transmission of neurotransmitter molecules [Faisal et al., 2008], to the trial-to-trial variability of neurons in cortex, to the muscular noise in movements [Harris and Wolpert, 1998]. This neuron behavior is therefore interesting for us, as it could be exploited as a source of slow stochastic drive in different neural systems. Note that achieving the same stochastic behavior with standard digital logic circuits might be resource consuming, as it would require the implementation of a random number generator.

3.3 Synapses

3.3.1 Spike-Timing Dependent Plasticity learning circuits

To endow large scale VLSI networks of spiking neurons with learning abilities it is important to develop compact and low power circuits that implement synaptic plasticity mechanisms. In recent years, there has been a proliferation of custom VLSI implementations of computing architectures based on spiking neural networks [Arthur et al., 2012; Bruederle et al., 2011; Choudhary et al., 2012; Giulioni et al., 2012; Indiveri et al., 2011; Yu et al., 2012]. These architectures compute in a distributed fashion that is massively parallel and fault tolerant and they can potentially have very low power consumption. Therefore, they are being explored as possible alternative solutions to conventional digital computing approaches. The "programming model" for these networks, however, is radically different from the conventional von Neumann sequence of instructions model. The knowledge, or the program, in a neural network is encoded in the network architecture, and in the weights of the connections between the neurons. New theories and methods are being developed for understanding how to synthesize pre-specified computing algorithms on these hardware systems through defining appropriate network architectures and connectivity patterns [Eliasmith et al., 2012; Neftci et al., 2013]. However, to endow these architectures with adaptation and learning mechanisms that change their synaptic weights on-line (e.g., to realize the intended functionality autonomously) it is necessary to develop appropriate compact and low-power plasticity circuits that are compatible with the VLSI architectures being developed. Furthermore, to allow these hardware neuromorphic systems to interact with the user and the environment (e.g., in the field of robotics or neuroprosthetics), it is important that these circuits can exhibit dynamics with biologically plausible time constants.

3.3.2 The spike-based learning algorithm

Many models of STDP have been proposed in the computational neuroscience literature [Abbott and Nelson, 2000; Markram et al., 2012]. However, a growing body of evidence is revealing that learning algorithms based on spike-timing alone cannot account for all of the phenomenology observed neurophysiological experiments [Lisman and Spruston, 2010], have poor memory retention performance [Billings and van Rossum, 2009], and require additional mechanisms to learn both spike-time correlations and mean firing rates in the input patterns [Senn, 2002]. For this reason, we chose to implement the spike-driven synaptic plasticity rule proposed by [Brader et al., 2007], which has been shown to reproduce many of the behaviors observed in biology, and has performance characteristics that make it competitive with the state-of-the-art machine learning

methods [Brader et al., 2007]. This algorithm does not rely on spike-timing alone. It updates the synaptic weights according to the timing of the pre-synaptic spike, the state of the post-synaptic neuron's membrane potential, and its recent spiking activity. It assumes that the synaptic weights are bounded, and that, on long time-scales, they converge to either a high state, or a low one. However, in order to avoid updating all synapses in exactly the same way, this algorithm requires a stochastic weight update mechanism (see [Brader et al., 2007] for details). The requirements and features of this algorithm make it particularly well suited for neuromorphic hardware implementation: the bi-stability feature removes the problematic need of storing precise analog variables on long-time scales, while the probabilistic weight update requirement can be obtained by simply exploiting the variability in the input spike trains (typically produced by a Poisson process) and the variability in the post-synaptic neuron's membrane potential (typically driven by noisy sensory inputs). The weight-update rule for a given synapse i is governed by the following equations, which are evaluated upon the arrival of each pre-synaptic spike:

$$\begin{cases} w_i = w_i + \Delta w^+ & \text{if } V_{mem}(t_{pre}) > \theta_{mem} \text{ and } \theta_1 < Ca(t_{pre}) < \theta_3 \\ w_i = w_i - \Delta w^- & \text{if } V_{mem}(t_{pre}) < \theta_{mem} \text{ and } \theta_1 < Ca(t_{pre}) < \theta_2 \end{cases} \tag{3.10}$$

where w_i represents an internal variable that encodes the bi-stale synaptic weight; the terms Δw^+ and Δw^- determine the amplitude of the variable instantaneous increases and decreases; $V_{mem}(t_{pre})$ represents the post-synaptic neuron's membrane potential at the time of arrival of the pre-synaptic spike, and θ_{mem} is a threshold term that determines whether the weight should be increased or decreased; the term $Ca(t_{pre})$ represents the post-synaptic neuron's Calcium concentration, which is proportional to the neuron's recent spiking activity, at the time of the pre-synaptic spike, while the terms θ_1, θ_2, and θ_3 are three thresholds that determine in which conditions the weights are allowed to be increased, decreased, or should not be updated. These "stop-learning" conditions are useful for normalizing the weights of all synapses afferent to the same neuron. They have been shown to be effective in extending the memory lifetime of recurrent spiking neural networks, and in increasing their capacity [Senn and Fusi, 2005]. In parallel to the instantaneous weight updates, the internal variable of the synapse w_i is constantly being driven toward one of two stable states, depending whether it is above or below a given threshold θ_w:

$$\begin{cases} \frac{d}{dt} w_i = +C_{drift} & \text{if } w_i > \theta_w \text{ and } w_i < w_{max} \\ \frac{d}{dt} w_i = -C_{drift} & \text{if } w_i < \theta_w \text{ and } w_i > w_{min} \end{cases} \tag{3.11}$$

where C_{drift} represents the rate at which the synapse is driven to its bounds, and w_{max} and w_{min} represent the high and low bounds respectively. The actual weight J_i of the synapse i is a thresholded version of the internal variable w_i that is used to produce the Excitatory Post Synaptic Current (EPSC) upon the arrival of the pre-synaptic spike:

$$J_i = J_{max} f(w_i, \theta_J) \tag{3.12}$$

where $f(x, \theta_J)$ can be a sigmoidal or hard-threshold function with threshold θ_J, and J_{max} is the maximum synaptic efficacy. We will show in Section 5.5 experimental results that demonstrate how the circuits integrated in the Neuromorphic Processor (NP) chip faithfully implement this learning algorithm.

3.3.3 The learning synapse circuit

In this paragraph, we describe a hybrid analog/digital weight-update circuit that can be biased to operate with biological realistic time constants. The circuit can be used in a wide variety of spike-based learning protocols and is optimized for minimum size and power-consumption. As the circuit updates its internal state variable when it is stimulated by input digital pulses, it is ideally suited to implement Spike-Timing Dependent Plasticity (STDP) learning algorithms. The synapse circuit updates its state in an analog way, continuously over time, but it settles to one of two possible states on long-time scales. This enables the design of an extremely compact and low-power circuit that is robust to noise in both the circuit and the input signals. The learning synapse circuit was integrated in a multi-neuron prototype chip, fabricated using a standard 180 nm CMOS process, comprising analog (adaptive exponential) integrate-and-fire neuron circuits, linear and non-linear synapse dynamics circuits, and digital asynchronous spike-based communication and processing circuits. Figure 3.5 shows the schematic diagram of the learning circuit. On the arrival of an input pulse spike_pre, representing a pre-synaptic spike, the circuit produces an output synaptic current I_syn that can be summed to the output currents of all other synapses, possibly fed through a linear integrator circuit that models synaptic dynamics, and conveyed to the post-synaptic integrate and fire neuron circuit. Multiple instances of these circuits can in turn be embedded in a network of spiking neurons for carrying out neural processing tasks. The amplitude of each pre-synaptic current I_syn depends on the corresponding circuit's synaptic weight. In the circuit of Fig. 3.5, the synaptic weight is encoded by the voltage Vc: if, during a spike, Vc<wthr! then the output current I_syn is zero (the synapse is in a Long-Term Depressed –LTD– state); conversely if Vc>wthr!, then the output current I_syn is equal to the Operational Transconductance Amplifier (OTA) bias current (the synapse is in its

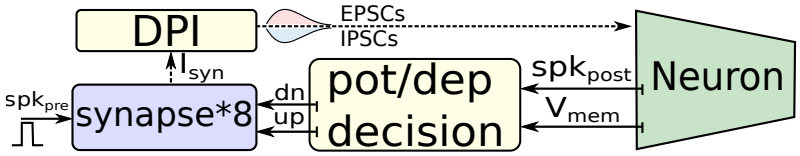


Figure 3.4: The learning architecture

Long-Term Potentiated –LTP– state). During (and only during) the spike, the OTA bias current is set by the wscale! analog bias. In the absence of spikes, the OTA is biased by a typically much smaller current, set by the drift! bias voltage, which slowly drives the Vc voltage towards the Vdd power rail if Vc>wthr! and towards Gnd if Vc<wthr!. The drift towards one of the two stable states can be extremely slow, and the power consumption can consequently be very small, because the OTA is configured in a positive-feedback configuration and it can operate correctly even with extremely small biases (e.g., using only leakage currents). The analog drift_up! and drift_dn! signals can be used to control the slopes of the up and down drifts independently. Learning takes place by updating the voltage Vc with each pre-synaptic spike: during a spike, the charge on the Vc capacitor can be either increased, decreased, or left unchanged, depending on the values of the control signals up and dn. The analog biases up_jump! and dn_jump! determine the amount of charge being dumped or sourced from the capacitor (and therefore the amplitude of the Vc change). The up and dn control signals can be shared among all synapses afferent to a neuron. Therefore only one circuit producing these signals needs to be instantiated (e.g., next to the neuron circuit). These signals can be generated based on post-synaptic spikes or based on the membrane potential of the postsynaptic neuron. We show in section 3.3.4 that generating the up and dn signals based only on the membrane potential is enough to reproduce the classical STDP learning algorithm [Abbott and Nelson, 2000]. However, more elaborate and powerful models can be implemented by using circuits that take into account additional postsynaptic variables [Lisman and Spruston, 2005]. For example, in the prototype chip that includes the synapse described, the circuits producing the up and dn signals are based on a spike-based learning algorithm originally described in [Brader et al., 2007], and successfully applied to pattern classification in neuromorphic hardware [Giulioni et al., 2012; Mitra et al., 2008].

Comparison to prior work

There are two circuits that have been presented in the past, which are closely related to the learning circuit we propose. The synaptic circuit described in [Badoni et al., 2006] has the same functionality as the synapse of Fig. 3.5: it generates either a high or low

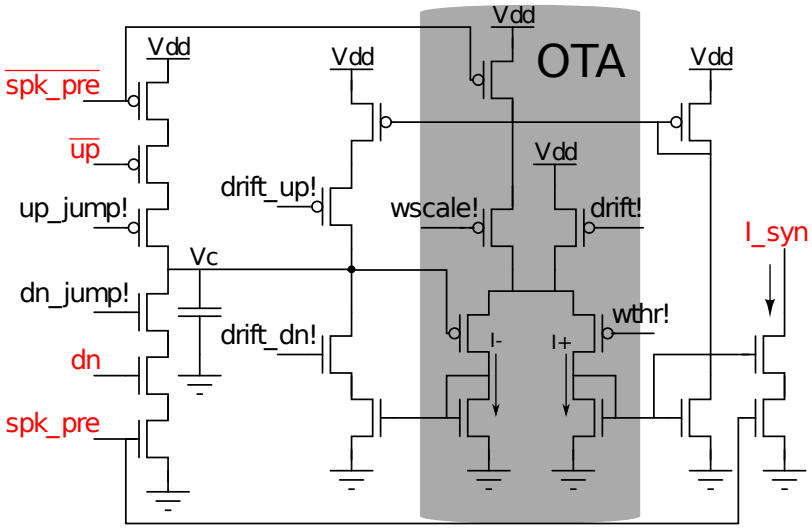


Figure 3.5: The low-power analog/digital synapse circuit. Pre-synaptic input spikes (spk_pre) trigger the weight update that can be positive or negative depending on the digital up and dn signals. Red text indicates port names. Net Names ending in "!" indicate externally supplied biases.

current, with every pre-synaptic spike, depending on the state of the internal synaptic weight variable, and drives the internal variable to a high or low state on longer time scales. However this circuit uses two distinct OTAs (one for the comparison, and one for the drift). In addition to being larger, the OTA performing the comparison between the internal variable and a threshold is constantly active and dissipates power continuously (as opposed to the circuit of Fig. 3.5 which activates this feature only during spikes, when it is needed). The other related learning circuit is the one that was originally described in [Indiveri and Fusi, 2007]. In this prior implementation the output synaptic current was not a threshold function of the internal variable Vc, but was directly proportional to the exponential of Vc (by means of a sub-threshold nFET). This was a more compact and low-power circuit, but with an extremely high sensitivity to weight fluctuations. We carried out experiments to highlight the differences between the two approaches (shown in Section 3.3.4), and point out the advantages of the new solution.

3.3.4 Synapse's circuit measurements

To show how the learning circuit of Fig. 3.5 is compatible with STDP learning rules, we generated a Poisson-distributed teacher spike train with a mean firing rate of 200 Hz sent via a fixed synapse. This teacher spike train causes the post-synaptic neuron to fire stochastically with a mean rate of 15 Hz. At the same time, we stimulated the learning

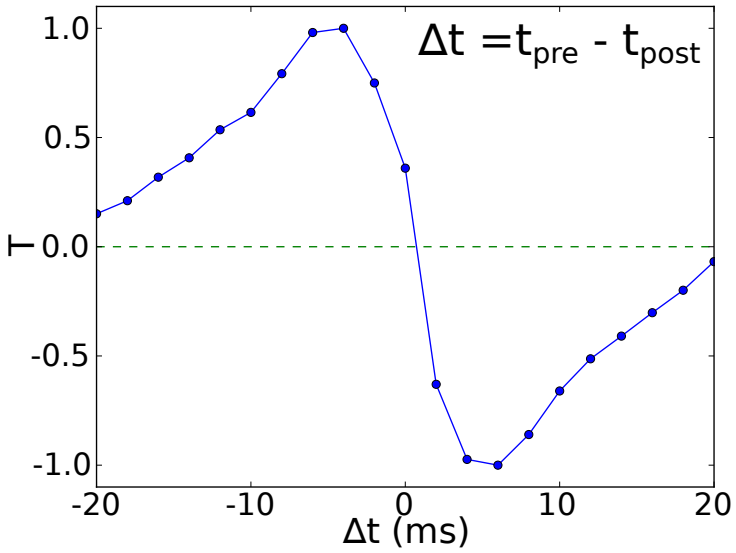


Figure 3.6: Spike-timing dependent transition tendency as a function of Δt between pre- and postsynaptic spikes.

synapse with a spike (pre-synaptic) shifted in time with respect to the post-synaptic spike and we observed the behavior of the circuit for different $\Delta t = t_{pre} - t_{post}$ spike times. Following the voltage-dependent weight update rule described in [Brader et al., 2007], our circuit exhibited higher probabilities of weight increase (Vc receiving the current generated through up_jump!) when pre-synaptic spikes preceded post-synaptic spikes. Similarly, the probability of weight decrease was larger for cases when post-synaptic spikes preceded pre-synaptic spikes. To quantify these effects, we computed the "tendency" function T, as defined in [Brader et al., 2007]:

$$T = \left(\frac{p_{up}}{p_{dn} + p_{up}} - \frac{1}{2} \right) \cdot max(p_{up}, p_{dn}) \tag{3.13}$$

where p_{up} and p_{dn} are the probabilities of up and down weight changes for a given timing difference Δt. The plot of this tendency T as a function of Δt is shown in Fig. 3.6. The figure shows that the learning circuits can reproduce the STDP behavior. In the next experiment, we measure the circuit's ability to realize slow temporal dynamics. This feature allows the network to process spike trains with low mean rates and biologically plausible temporal dynamics. Figure 3.7 illustrates the slow dynamics of the synaptic weight when it is drifting in the absence of presynaptic spikes. The drift! and the

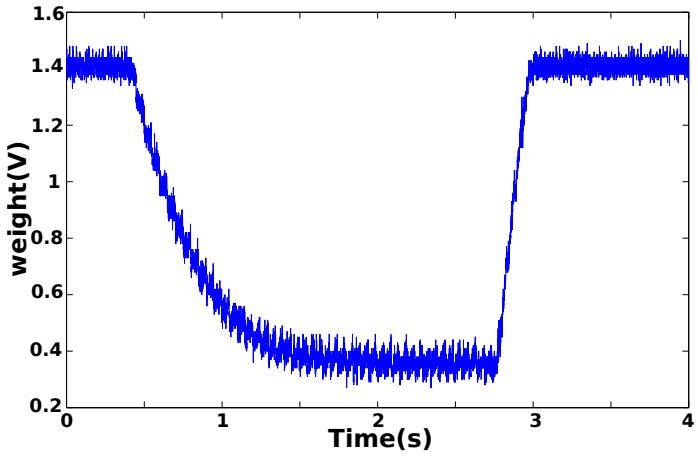


Figure 3.7: Slow drift of the synaptic weight. The biases were changed at 0.4s and at 2.7s to trigger down and up drifts respectively.

drift_up! bias voltages of Fig. 3.5 were both set to 1.8V, drift_dn! was set to 0.02V. wthr! was switched from 0V to 1.8V at 0.4s and back to 0V at 2.7s. Using such slow drift rate, weight updates can be sensitive to temporal correlations at the 1s scale in the presynaptic spike train. In the experiment of Fig. 3.8, we compare the effect of learning in the synapse proposed in this work with the synapse previously proposed in [Indiveri and Fusi, 2007]. Measurements were carried out on two separate chips that implement the two versions of the circuit. A neuron in each chip was stimulated via the plastic synapse. We set the synaptic weights to their high states in both circuits, set the biases so as to induce downward jumps in the Vc nodes, and stimulated the synapses with regular 200 Hz pre-synaptic spike trains. The stimulation starts at $t = 0.05$ s. The top plot shows the measurements from the prior implementation: even a very small reduction in the internal synaptic weight variable drastically alters the current injected into the neuron. As soon as the weight drops slightly, the synaptic output current becomes so small that it is not able to drive the neuron anymore. In contrast to this behavior, the bottom plot shows that changes in the internal variable Vc (labeled as W in the figure legend) have an effect only if they make it cross the threshold bias wthr! of Fig. 3.5 (labeled as W_{thr} in Fig. 3.8). Therefore this synapse is robust to small fluctuations in its synaptic weight internal variable (e.g., induced by noisy input spikes), and consolidates the changes that it is required to learn only if there are enough consistent signals that drive it to make a transition (note the drift component in the W traces of Fig. 3.8). We investigated the effect of the presynaptic weight on the neuron firing rate. The firing rate of a neuron driven by a regular spike train through the plastic synapse only changes

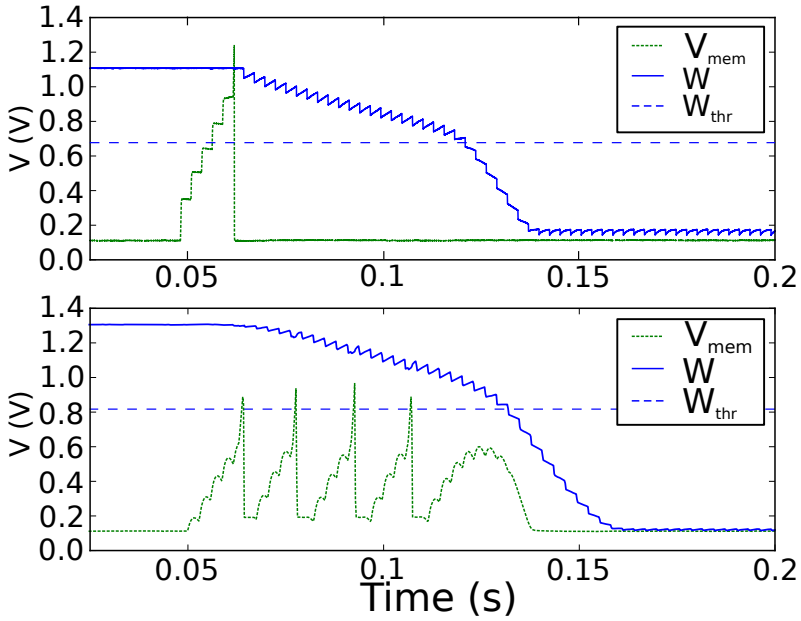


Figure 3.8: Effect of fluctuations in the synaptic weight on the response of the postsynaptic neuron. The top plot shows measurements from the prior learning circuit implementation. The bottom plot shows measurements from the current implementation. Synaptic weights are actively driven to a low state. When the weights W are above the threshold W_{thr}, the down jumps act against a slow drift towards the high state. As soon as the weights cross W_{thr}, the drift acts in the same direction of the down jumps.

when the synaptic weight crosses the bi-stability threshold wthr!.

3.4 Address Event Representation (AER)

Neuromorphic event-based systems consist of multiple custom hybrid analog/digital Very Large Scale Integration modules that communicate among each other using the Address Event Representation, which was first introduced by [Boahen, 2000; Lazzaro et al., 1993]. In this representation each source or sender node (e.g., a silicon neuron or a more in general a pixel) is assigned an address, and when it produces an event (e.g., a spike) its address is instantaneously put on a digital bus, using asynchronous logic. Destination nodes (e.g., synapses) can decode and consume these asynchronous address-events at the time in which they receive them. In the work of this thesis we use a single-sender/single-receiver communication scheme, a handshaking mechanism ensures that all events generated at the sender side arrive at the receiver. Signals are encoded using a Bundled Data (BD) representation, in which the address of the sending element

is conveyed as a parallel word of sufficient length, and two additional lines are required for the handshaking control signals. Systems containing more than two AER modules are constructed by implementing additional purpose arbitration schemes [Imam et al., 2012; Jin et al., 2010; Merolla et al., 2007; Serrano-Gotarredona et al., 2009]. In AER systems therefore time represents itself, input and output address-events are transmitted using asynchronous digital pulses that encode the address of the sending node, and analog information is carried in the temporal structure of the inter-pulse intervals and in their mean frequency. If multiple senders generate events simultaneously, an arbitration scheme makes sure that the addresses do not collide, but are transmitted on the bus in sequence. The arbiters that manage event-collisions are implemented using asynchronous digital logic circuits. These asynchronous digital circuits change the state of their memory elements in response to transitions on the data lines. This is very different from what happens in *synchronous* logic, where all state transitions occur at the edges of a global clock. Synchronous logic is more common than asynchronous logic by far. Indeed the predominance of synchronous logic has led to powerful Electronic Design Automation (EDA) tools that greatly accelerate the design process. While there have been recent advances in the research and design of asynchronous logic, especially for neuromorphic systems [Imam et al., 2012; Jin et al., 2010; Merolla et al., 2007], there is still no mature and readily available digital design flow for the development and automated design of asynchronous circuits.

3.5 Asynchronous event-based current to frequency converter

Analog to digital conversion of currents in the range $10^{-12} \leq I \leq 10^{-5}A$ need to be performed in order to measure I_{mem} and I_{ahp}. These signals are the crucial internal variables of neuron's soma circuit. In order to access these variables externally I designed and fabricated an asynchronous event-based analog-to-digital converter. The use of an analog-to-digital converter is mandatory for such small currents as they could get dissipated, if not previously amplified, in the passage though the pad-frame of a VLSI chip. This is caused by the Electro Static Discharge (ESD) protection circuits necessary to protect the delicate asic circuits. The schematic of the converter is shown in Fig. 3.9. The main idea is to convert the current I_{in} into a frequency signal. In fact in both schematic diagrams a current (positive or negative) is used to charge a capacitor that integrates it and whose voltage level drives a chain of two inverters. When the first inverter is switched on a pulse is emitted at the REQ terminal and an ACK signal discharge the capacitor resetting the circuit for the next measure. The ADC pulse-output circuit is an asynchronous clock-less circuit that is based on the same principle of the Integrate-and-Fire (I&F) neuron; it produces spikes at a rate that is proportional to

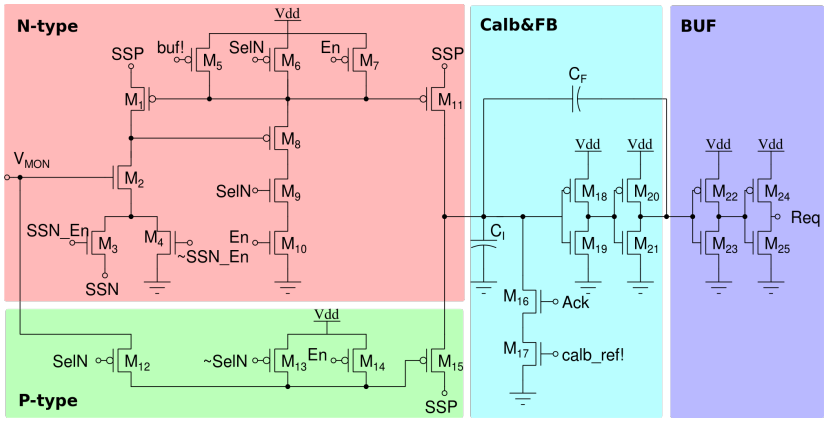
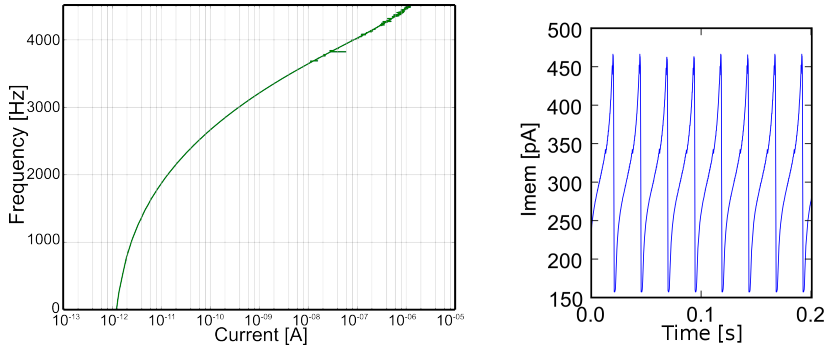


Figure 3.9: Current to frequency converter.



(a) (b)

Figure 3.10: 3.10a) current to frequency converter characteristics. 3.10b) ADC output of somatic current.

the intensity of its input. The circuit contains two programmable branches that enable the selection of current direction (p-type 3.9) or (n-type 3.9). In Fig. 3.10 we show a reconstructed somatic current from the output this circuit together with the measured current-frequncy characteristic of the circuit. The measure has been obtained with the help of an FPGA whose role was to count number of events at fixed time bins ($t_{bin} = 200ms$). The sum of events in every bin has been the converted to a current value by using the characteristic of the converter circuit 3.10a.

3.6 Discussion and Conclusions

We designed, fabricated and tested novel circuits for implementing neuron and synapse dynamics. We introduced a new bi-stable synapse, characterized their properties with

experimental measurements, and showed the improvements made over previous implementations. The synapse proposed is extremely compact and low power, at the cost of having only two stable states as possible synaptic weights. The choice of implementing binary synapses with analog circuits may seem strange. However, it has been demonstrated that, despite the limited resolution, binary synapses are sufficient to learn random uncorrelated patterns of spike-trains, as in the perceptron case: perceptrons that use binary, stochastic synapses have been shown to work in software and hardware tasks in the past [Badoni et al., 2006; Brader et al., 2007; Giulioni et al., 2012; Mitra et al., 2009; Senn and Fusi, 2005], and represent a good compromise between implementation complexity and functionality. In order to allow the realization of on-line linear classifiers with these synapses, it is important to use stochastic learning paradigms [Senn and Fusi, 2005] and realize circuits where the synaptic efficacy is affected by the neuron only when the synaptic state crosses the bi-stability threshold [Badoni et al., 2006]. The bi-stability threshold bias (`wthr!` of Fig. 3.5) can further be used to change the balance between potentiation or depression without affecting the dynamics of the post-synaptic neuron, for example to realize reward-modulated learning or to compensate for circuit inhomogeneities due to mismatch. The variability embedded in input (or natural) spike trains can be effectively used as a source of stochasticity for implementing stochastic learning without requiring additional sources of noise or random number generators. Variability or noise in the input will not degrade performance: only when a sufficient number of spikes consistently impose potentiation or depression on a synapse, its internal state will counteract the drift current and undergo an LTP or LTD transition respectively (e.g., see LTD transition in Fig. 3.8). These transition events are rare and depend on both the pre-synaptic firing rate and on the dynamics of the post-synaptic neuron. The circuits we proposed can be used to realize compact arrays of distributed learning synapses because only local information, i.e., the post-synaptic membrane potential, is used to realize the STDP. Alternative implementations instead require mechanisms to store of spike-timing differences, computation the corresponding synaptic weight changes and deliver them to the synapses [Pfeil et al., 2012; Seo et al., 2011]. Networks of silicon neurons that have these types of synapses and that use stochastic learning can be used effectively as an ensemble of linear classifiers trained with supervised learning to solve non-linear classification tasks. In this way, synapses corresponding to the same inputs connected to different post-synaptic neurons will undergo independent stochastic processes. Hence, even though the neurons in the ensemble of classifiers are trained with the same data, the resulting classifiers will have slightly different responses. By aggregating the responses of these neurons, a single, optimal, classifier can be realized in a single-layer neural network using the bi-stable synapses and the circuits we proposed [Senn and Fusi, 2005].

This strategy is analogous to bootstrap aggregation methods in machine learning where multiple classifiers are independently trained and their responses combined [Breiman, 1996]. Furthermore, by allowing only a subset of synapses to make synaptic transitions upon the presentation of the patterns to be learned, the system "samples" from a distribution of independent classifiers, with the result that the learned perceptron with bi-stable synapses will have better performance than a deterministic one [Hinton et al., 2012]. There are two main sources of stochasticity that the bi-stable synapses employ, one associated with the pre-synaptic and one with post-synaptic spike-train. In the assumption of uncorrelated pre-synaptic spike-trains, two synapses of the same neuron will undergo different long-term transitions. On the other hand, spike-timing correlations of the pre-synaptic spikes would introduce strong correlations on the synaptic state changes, since the state changes for both synapses are governed by the same post-synaptic activity. For example, two (ideal) synapses receiving coincident spikes would behave in the same manner. Consequently, the speed at which the patterns are stored can be modulated by the correlations of the pre-synaptic spike-trains. Since synchronicity in the spike-trains is often associated with attentional mechanisms [Niebur, 2002; Steinmetz et al., 2000], this effect could be used for attention-modulated learning. In this paper we have shown how correlations in the spike-timing of the pre- and post-synaptic activities can modulate the probabilities of synaptic transitions by directly measuring these probabilities.

CHAPTER 4

Spikebetter: a programmable neural core processor with an asynchronous SRAM for enhanced synaptic weights precision

F. Corradi, H. Mostafa, M. Osswald, S. Moradi, and G. Indiveri

We developed a neuromorphic programmable core in which neural computation is performed in the analog domain by an array of 58 neurons. Every neuron is connected to a dendritic arbor of 8 binary analog Hebbian-like synapses that implement Spike–timing–dependent Plasticity. These synapses can undergo conformational changes (i.e. Long Term Potentiation (LTP) and/or Long Term Depression (LTD)) depending on the spiking activity of pre- and post-synaptic neurons. Along with learning synapses every neuron is connected to 32x4*bits* fixed-weight programmable synapses implemented in a custom asynchronous SRAM architecture. The communication of action potentials and the storage of synaptic weights is performed using custom digital asynchronous logic. The neural dynamics are implemented in the analog circuits that instantiate adaptive-exponential integrate-and-fire neurons with biologically realistic synaptic dynamics in a very compact and power efficient way. The system is capable of matching the time constant of real-world stimuli as well as the time constant of the corresponding neuro-biological sensory systems. The custom digital circuits implement a real-time event based communication protocol following the AER standard [Lazzaro et al., 1993]. The full architecture implements a feed-forward scheme, spikes are presented at the asynchronous input AER logic, their address is decoded and they are directed to a single synapse in the array. Neurons integrate synaptic current contributions and and emit a spike when

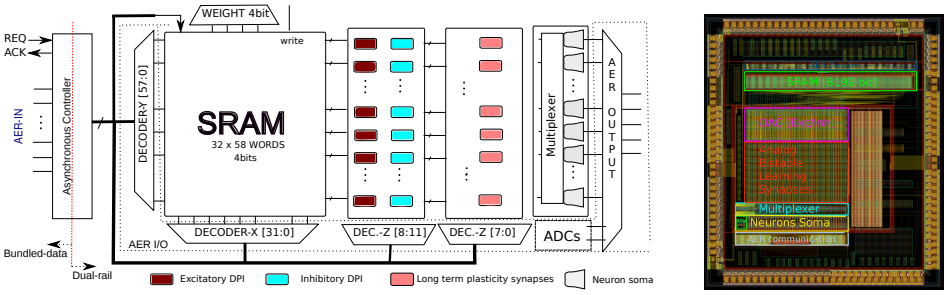
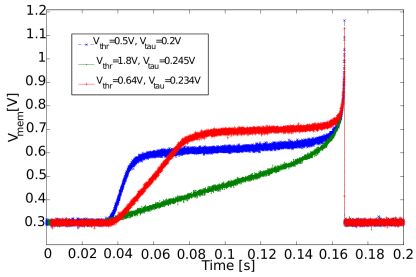


Figure 4.1: Spikebetter neuromorphic multi-neuron chip architecture

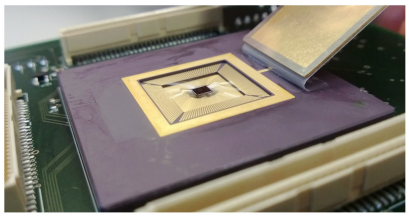
their membrane potential crosses a threshold. The spike will appear on the AER output logic block. Input and output interfaces follow the four-phase handshake protocol, and the addresses are presented in a parallel bit configuration at the input and output pins.

4.1 Chip architecture

The architecture of the chip is illustrated in figure 4.1. The device has been fabricated using a standard 0.18 μm CMOS process and it occupies an area of approximatively $3mm^2$. The chip architecture comprises five main blocks: the asynchronous encoder and decoder, SRAM block, the Digital to Analog Converters, a learning synapses block and a neuron core. The AER block is an asynchronous digital communication circuit with four phase handshake [Mostafa et al., 2013]. In this chip we implemented 58 adaptive exponential integrate-and-fire neurons. Each neuron has a dendritic tree that comprises up to 32 SRAM cells and 8 learning synapses. The asynchronous SRAM block is used to store synaptic weight values with 4 bit resolution. This memory includes a dual-rail representation of the data which exploits a filter circuit first introduced in [Ekanayake and Manohar, 2003] and also used in a previous neuromorphic core [Moradi and Indiveri, 2014]. The digitally stored synaptic weights are converted into analog currents by the Digital to Analog Converter (DAC) block. This block produces positive or negative currents to simulate excitatory or inhibitory post-synaptic currents. Since the filter circuits implemented in the synapse DAC integrators are linear it would be sufficient to use a single synapse per neuron and multiplex it in time. In this way one could represent n synapses with a single memory cell. However in this scheme it would not be possible to obtain excitatory and inhibitory synapses, as well as different time constants. In addition to the programmable SRAM synapses there is a block of 8 learning synapses as the one



(a)



(b)

Figure 4.2: 4.2a. Membrane potential for different bias voltage parameters. 4.2b. Photo of the neuromorphic chip.

presented in this thesis in Section 3.3.1.

4.2 Neurons array

The neuron circuit is the adaptive exponential integrate-and-fire circuit previously described in Section 3.2.1 and shown in figure 4.3. We refer to that section for circuit description and behavior analysis. Here we only report that this circuit has been demonstrated to be extremely low-power while being enable to express different patterns of neural activity as Class I/II excitability patterns, as well as bursting behavior and spike-frequency adaptation (see Section 3.2.1 for details). The power consumption of a similar implementation [Livi and Indiveri, 2009] has been reported to be 7 pJ per spike. Figure 4.2a shows a neuron's membrane potential trace.

4.2.1 Neuron membrane potential

The dynamics of the membrane potential can be described by a two-variables system of differential equations, as in [Brette and Gerstner, 2005]. The first equation describes the activation function with an exponential voltage dependence. The membrane voltage is then coupled to a second equation which describes the adaptation of the neuron:

$$\begin{aligned} C\frac{dV}{dt} &= -g_l(V - E_l) + g_l\Delta_t exp\left(\frac{V - V_t}{\Delta_t}\right) - w + I_{bias} \\ \tau_w\frac{dw}{dt} &= a(V - E_l) - w, \end{aligned} \tag{4.1}$$

The neuron membrane potential is V, w is the adaptation variable, I_{bias} is a constant input current, C is the membrane capacitance, g_l is the leak conductance, E_l is the leak reversal potential, V_t is the threshold, Δ_t is the slope factor, a is the adaptation coupling parameter, and τ_w is the adaptation time constant. The synaptic efficacies are described

by the following equations:

$$\begin{aligned} \frac{dg_e}{dt} &= -\frac{g_e}{\tau_e}\sum_n \delta(t - t_{in}), \\ \frac{dg_i}{dt} &= -\frac{g_i}{\tau_i}\sum_n \delta(t - t_{in}), \end{aligned} \tag{4.2}$$

g_e is the conductance for excitatory synapses, g_i is the conductance for inhibitory synapses, τ_e is the time constant for excitatory synapses, τ_i is the time constant for inhibitory synapses, and δ represents the arrive of a spike at time t_{in}. Figure. 4.2a shows three traces that represent the membrane potential for three different bias threshold voltages V_{thr} and integration time constants V_{tau}. The neuron circuit is shown in Figure 4.3. The synaptic input current I_{syn} is low-pass filtered by a Differential Pair Integrator (DPI) filter [Bartolozzi and Indiveri, 2007a] (see Fig. 4.3a), which includes an adjustable threshold voltage transistor V_{thr} and a leak conductance bias V_{tau}. The membrane capacitance integrates the input current and generates the membrane potential V_{mem}. A positive-feedback inverting amplifier is used to generate spike events with very low-power consumption [Indiveri et al., 2011] (see Fig. 4.3c). Spike frequency adaptation is implemented by an additional DPI circuit in negative feedback configuration (see Fig. 4.3b). The magnitude of the adaptation current can be controlled via the bias voltage V_{adw}, while the time constant is controlled by the voltage bias V_{adtau}. Figure 4.3d contains the spike communication inverter (M_{D4}, M_{D3}), the refractory period bias V_{refr} transistor and the bias that regulates the reset voltage V_{reset} of the neuron after a spike event.

4.2.2 Single neuron activation function

The activation function of a neuron shows the dependence between mean firing activity of the neuron and its presynaptic input. Neuronal parameters have been chosen such that there would be no mean difference in the neuron activation function. We present a measure of the single neuron activation function in Fig. 4.4. The shaded areas are the standard deviation of the measure. The measure has been obtained by stimulating all neurons in the two populations with homogeneous Poisson spike trains of various frequencies (x-axis). It is important to note that all neurons were disconnected from each other. The activation function presents a linear regime of excitability $50 < \nu_{in} < 100$. For $\nu_{in} > 100$, the excitability starts to decrease. The different slope is due to the saturation caused by the refractory period.

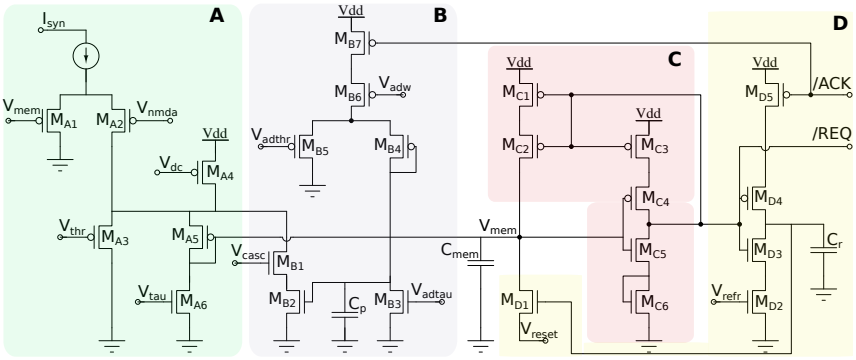


Figure 4.3: Adaptive exponential I&F neuron circuit schematic. (a) Current input block. (b) Adaptation circuit. (c) Positive-feedback inverting amplifier. (d) Digital communication and reset block.

4.3 SRAM architecture and cell considerations

The architecture of the asynchronous Static Random Access Memory block is shown in figure 4.6. It is composed of a $10T$ memory cell and a transmission gate stage which enables the output of the cell upon a request (PU) signal. There is an additional filter circuit stage that produces dual-rail data representation. We decided to implement the memory cell with a $10T$ configuration because this cell offers a better memory retention than a standard $6T$ memory cell. It also provides higher threshold for the read and write operations compared to the standard $6T$ cell, as shown in figure 4.3.1. $ST2$ cell performs close to $ST1$ cell in write and read trip points (see. Fig. 4.3.1) but this cell requires an additional control signal $WL1$. For these reasons we decided to implement the Static Random Access Memory block with a $10T, ST1$ memory cells. In figure 4.8 we show the layout of the cell, as well as its tiling. This cell occupies an area of $20.2um^2$ in a conservative Complementary Metal–Oxide–Semiconductor $1P6M$ $180nm$ technology.

4.3.1 SRAM operations

- *Idle* : when no input is present the Bit and nBit signals are actively pulled up to VDD, therefore the output of the filter circuits are both set to GND
- *Read* : when the decoder-X selects a column through the WL line, the $4-bits$ contents of the four selected memory cells are presented at the ouput of the memory cells. At the same time the decoder-Y enables the transmission gate block, allowing the bit memory content (Bit,nBit) to arrive at the input of the filter circuit. This filter circuit generates a dual rail representation of the data which represents the memory content.

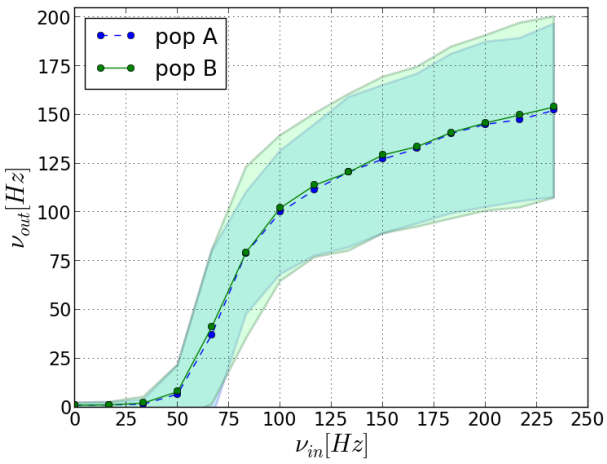


Figure 4.4: **Single neuron activation function.** Shaded areas represent the standard deviation of the measure. Blue dots refer to the mean firing rate activity for neurons in population B. Green dots refers to mean firing rate activity for neurons in population A.

- *Write* : the memory is programmed by setting the $Write - enable$ signal to high, together with the transmission of a single Address-Event (AE) containing the address and the content of the memory word. Since only a single memory block of four bits can be addressed, only a single memory word can be programmed with an AE. However, the content of the memory word is also passed though and directed to the DAC which will generates currents and excite the neural core. This allows users to both store and set synaptic weights on-line.

4.4 The synapse Digital to Analog Converter block

Synaptic weights are stored with 4-bit resolution in a standard 10T SRAM block [Lo and Huang, 2011] integrated on the same chip with an asynchronous interface. The digital weight values are converted into an analog current by a DAC circuit integrated in the DPI synapse circuits (see Figure 4.9). The bias voltages W_0, W_1, W_2, and W_3 are used to tune the current of the single bit value. The *PU*-biased p-Field Effect Transistor (FET) is necessary to eliminate charge-pump effects. The DAC output current $I_{synSRAM}$ determines the gain of the excitatory or inhibitory DPI synapse. The excitatory (inhibitory) synapse produces an exponential EPSC (Inhibitory Post-Synaptic Current (IPSC)) which emulates the exponential ligand-gated postsynaptic current generation mechanism [Bartolozzi and Indiveri, 2007a]. When the output of the

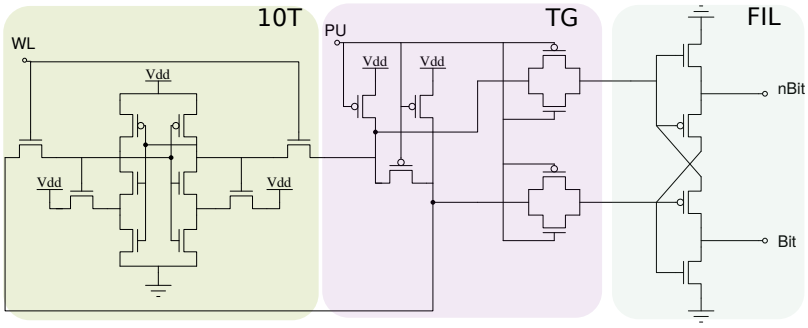


Figure 4.5: Asynchronous Static Random Access Memory architecture. It is composed of a $10T$ memory cell, a transmission gates stage, an output filter circuit to produce dual-rail data representation.

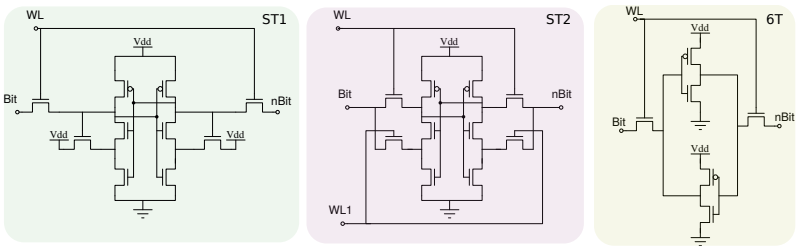


Figure 4.6: Three Static Random Access Memory cells, two $10T$ cells and the standard $6T$ cell. $ST1$ memory cell relies on a single control signal WL as the $6T$ while offering better scaling performance at a price of 4 additional transistors (see Fig. 4.3.1).

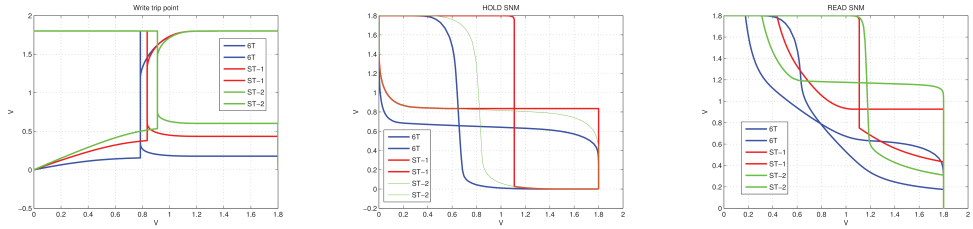


Figure 4.7: Noise margin cadence simulations

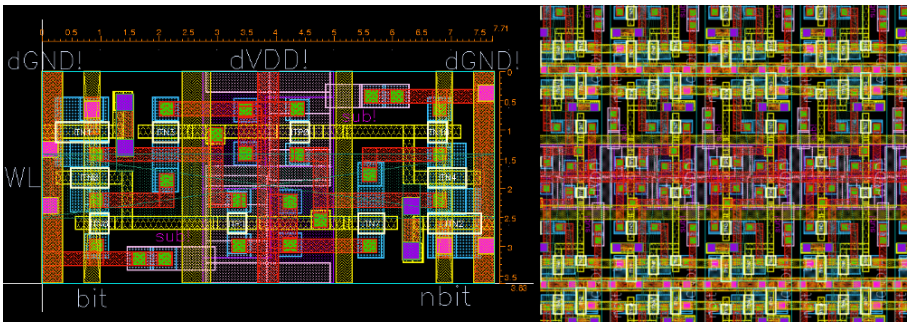


Figure 4.8: Static Random Access Memory layout and tiling

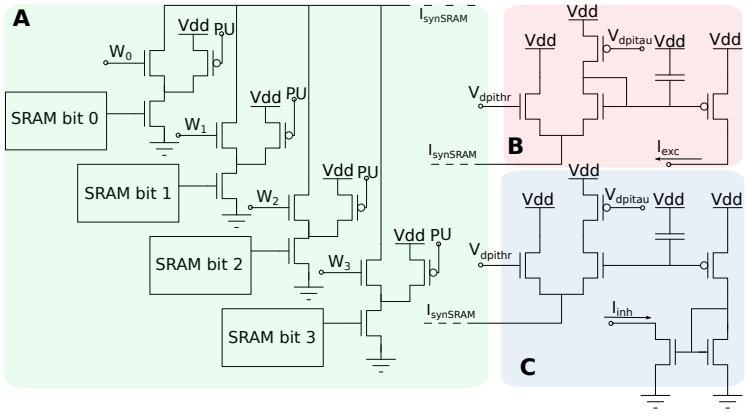


Figure 4.9: Programmable synapse circuit. (a) SRAM digital to analog converter block. (b) Excitatory DPI synapse. (c) Inhibitory DPI synapse.

asynchronous Static Random Access Memory memory block produces a valid dual-rail representation data, the synapse validity-check raises an *ack* signal and feeds the memory data to the Digital to Analog Converter block. Since the *ack* signals of all synapses are wired in a big logical OR block, the result is used by the asynchronous controller to produce a valid four-phase handshake cycle. The address-event data sent to the decoder-Z (see Fig. 4.1) is used to select different variants of excitatory or inhibitory Differential Pair Integrator circuits. The asynchronous logic paths of the SRAM and the neural-core blocks are independent. For correct operation, the decoder-X output should be ready before the weight bits are sent to the synapse's DAC. We assume that this is true, because we make the timing assumption that the decoder-X data path is faster than the memory access-time. The memory access time includes both the decoding time and the time required for the bitline signals to be driven by the memory control circuits.

4.5 Synapse's learning circuit array

The synapse's learning circuit implements a bistable Hebbian-like learning rule [Fusi, 2003; Fusi et al., 2000]. The core of the learning circuit has been described in Section 3.3.1 [Mostafa et al., 2014]. The synapse's circuit behaves as an integrator on short time scales with the preservation of the memory as one of two stable states (efficacies) on long time-scales. In the current section we focus on the system level circuits description, for implementing arrays of learning synapses in the spikebetter neuromorphic processor. The diagram of the analog/digital weight storage and update circuit is shown in Fig. 4.10. These circuits are divided into four blocks: the SET block can be used to set/reset the bistable state of the synaptic weight by sending an AER event with the matching address and properly asserting the configuration signals set_hi and set_low. The JUMP block increases or decreases the synaptic weight internal variable (i.e., the voltage V_w) depending on the digital signals up and dn, that are buffered copies of the ones generated in the silicon neuron stop-learning block (see Section 5.3). The heights of the up and down jumps can be set by changing the delta_up! and delta_dn! signals. The BIST block consists of a wide-range transconductance amplifier configured in positive feedback mode, to constantly compare the V_w node with the threshold bi_thr!: if $V_w >$ bi_thr! then the amplifier slowly drives the V_w node towards the positive rail, otherwise it actively drives it towards ground. The drift rates to the two states can be tuned by biases drift_up! and drift_dn! respectively. The current converter (CC) block converts the V_w voltage into a thresholded EPSC with maximum amplitude set by pa_wht!. In addition to the learning synapse circuit, it is necessary to add a circuit block that evaluates the weight update and "stop-learning" conditions. These circuits integrate the spikes produced by the post-synaptic neuron into a current that models the neuron's Calcium concentration,

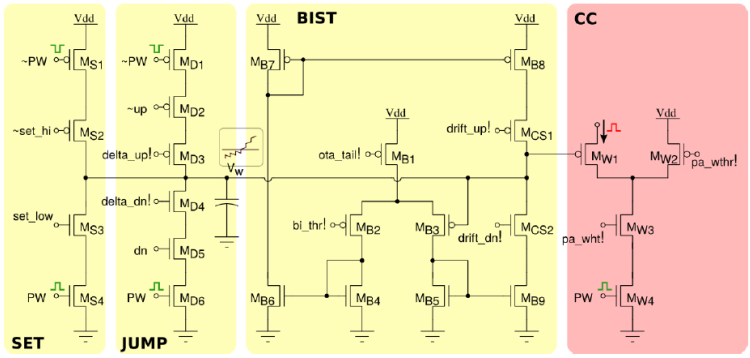


Figure 4.10: Long-term plasticity synapse schematic diagram.

and compares this current to three threshold currents. In parallel, the neuron's membrane current is compared to an additional threshold.

The schematic diagram of this circuit is shown in Fig. 4.11. The DPI M_{D1-D5}; computes the post-synaptic neuron's Calcium concentration; the comparisons with the fixed thresholds are made using three current-mode Winner-Take-All (WTA) circuits M_{W1-W9}, $M_{WU1-WU12}$, and $M_{WD1-WD12}$. The digital outcomes of these comparisons set the signals slnup and sldn which are then buffered and transmitted in parallel to all synapses afferent to this neuron belonging to the long-term plasticity array.

4.5.1 Long Term Potentiation and Long Term Depression probabilities measurements

To show the match between theory [Fusi, 2003; Fusi et al., 2000] and circuit behaviour we present a measure of LTP and LTD probabilities, for a single set of parameters of the synaptic dynamics, as a function of the spike rates of pre and post-synaptic neurons. The results of our experiments shows that we can fine-tune the transition probabilities and achieve a balance between them via the manipulation of the control bias currents. In a real-world situation the LTP and LTD transitions are stochastic as the number and distribution of events in any finite interval fluctuates. We initialize the state of the learning synapses with a binary flat random distribution. We generate Poisson spike trains for the presynaptic neuron using a standard workstation. The neuron's soma, excited by another Poisson input directed to a set of fixed weight SRAM synapses, emits spikes and the distribution of depolarization is used to generate the UP and DN jump probabilities $Q_a(Q_b)$ for the post synaptic neuron to be found above (below) the threshold θ_v. The probability Q_b or $1 - Q_a$ is the probability that the depolarization of the postsynaptic neuron is below the threshold θ_v. This probability indirectly relates to

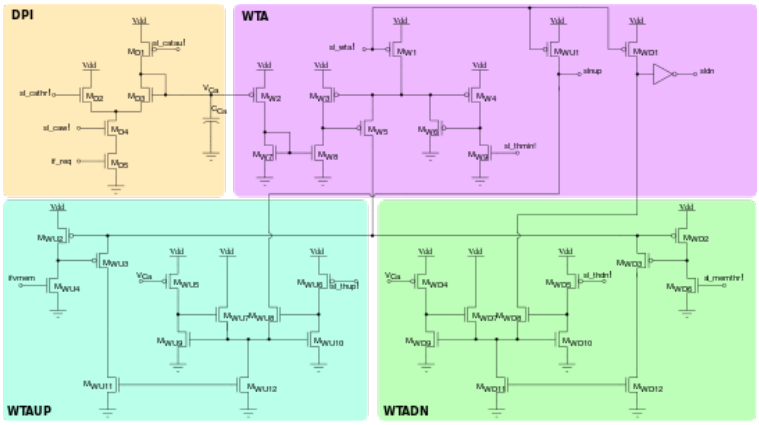
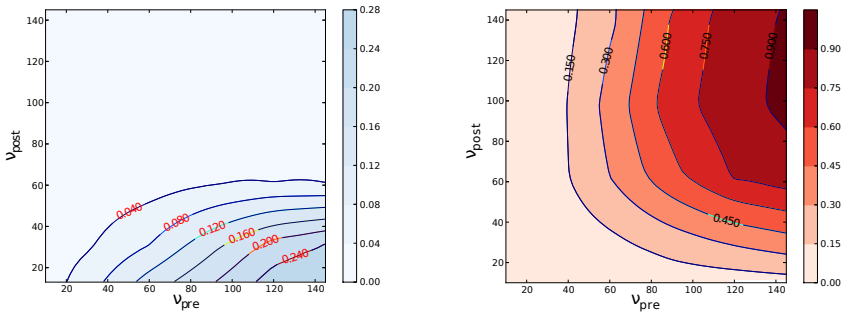


Figure 4.11: Post-synaptic learning circuits for evaluating the algorithm's weight update and "stop-learning" conditions. The DPI circuit M_{D1-5} integrates the post-synaptic neuron spikes and produces a current proportional to the neuron's Calcium concentration. Three current-mode winner-take-all circuits WTA, $WTAUP$, and $WTADN$ compare the Calcium concentration current to three set thresholds sl_thmin! , sl_thdn!, and sl_thup!, while the neuron's membrane current is compared to the threshold sl_memthr!.

the firing rate of the postsynaptic neuron, depending on the neural dynamics and the structure of the input current [Fusi et al., 2000]. Following the stimulation, the host workstation read the learning state of all synapses and establish whether a transition has occurred. This procedure is repeated for a range of presynaptic and postsynaptic frequencies. Figures 4.12a and 4.12b show the results of the experiment. The probabilities of transition are color coded; on the x-axis are the mean presynaptic input frequencies and on the y-axis are the postsynaptic frequencies. From Fig. 4.12a we see that when the presynaptic neuron is firing at high frequency ($> 60Hz$), while the postsynaptic neuron is firing at low frequencies ($< 60Hz$) the probability of LTD is high, indicating an anti-causal pre-post stimulus pair. On the other hand by examining Fig. 4.12b we see an increase in LTP probability for an increase in both pre and postsynaptic frequencies, indicating a causal paired stimulation.

4.6 Discussion and Conclusions

The control we have over the values of synaptic weights can be used to implement different models of VLSI spiking neural networks. Many solutions have been used [Choudhary et al., 2012; Rast et al., 2008] to provide this ability in hardware implementation of spiking neural networks. FPGAs also offer a flexible environment to realize programmable and/or probabilistic synaptic weights, as in [Choudhary et al., 2012]. However these



(a) LTD (b) LTP

Figure 4.12: 4.12a: Long-Term-Depression probability. 4.12b: Long-Term-Potentiation probability

solutions are not suitable for low-power and embedded applications mainly because of area and power constraints. These aspects of the problem have been considered during the design of Spikebetter. The use of analog sub-threshold techniques shows promising area and power efficiency factors for extending the design to large scale systems. On the other hand, since the design is composed of mixed signal anlog/digital circuits such as as the DAC and the neurons that work in sub-threshold, calibration and parameter tuning is a key issue while mapping spiking neural networks into these types of devices. In our solution we exploited compact SRAM cells and fast digital asynchronous logic to either program the synaptic weight on-line with an incoming Address-Events, or to compute weights off-line and store their values in the SRAM. The synapse output currents, after conversion with the DAC, are integrated using low-power and efficient adaptive exponential integrate-and-fire silicon neuron, with programmable refractory period, and spike-reset voltage. In Chapter 8 I demonstrate how to exploit this hardware for mapping spiking neural network for computing arbitrary mathematical functions and dynamical systems. The approach is based on an off-line weight optimization technique and on a parallel calibration procedure for setting synaptic weights. In conclusion, we designed, fabricated, and tested a neuromorphic processor that comprises 58 adaptive exponential integrate-and-fire neurons, 2K programmable SRAM synapses, and 464 plastic bi-stable Hebbian-like synapses. The chip is a feed forward chip in which spikes are communicated via the asynchronous input digital interface that follows a four-phase handshake protocol. These AEs can be of two type: programming events or spiking events. The system is meant to implement many different spiking neural networks that can be trained off-line and downloaded into the SRAM block. The system also allows the implementation of networks in which synaptic weights change on-line, thanks to the learning synapses.

These synapses are few compared to the SRAM synapses, as they require more area.

CHAPTER 5

A Re-configurable On-line Learning Spiking Neuromorphic Processor comprising 256 neurons and 128K synapses

N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri

Frontiers in Neuroscience, 2015

Implementing compact, low-power artificial neural processing systems with real-time on-line learning abilities is still an open challenge. In this work we present a full-custom mixed-signal VLSI device with neuromorphic learning circuits that emulate the biophysics of real spiking neurons and dynamic synapses for exploring the properties of computational neuroscience models and for building brain-inspired computing systems. The proposed architecture allows the on-chip configuration of a wide range of network connectivities, including recurrent and deep networks, with short-term and long-term plasticity. The device comprises 128 K analog synapse and 256 neuron circuits with biologically plausible dynamics and bi-stable spike-based plasticity mechanisms that endow it with on-line learning abilities. In addition to the analog circuits, the device comprises also asynchronous digital logic circuits for setting different synapse and neuron properties as well as different network configurations. This prototype device, fabricated using a 180 nm 1P6M CMOS process, occupies an area of 51.4 mm^2, and consumes approximately 4 mW for typical experiments, for example involving attractor networks. Here we describe the details of the overall architecture and of the individual circuits and present experimental results that showcase its potential. By supporting a wide range of cortical-like computational modules comprising plasticity mechanisms, this device will

enable the realization of intelligent autonomous systems with on-line learning capabilities.

5.1 Introduction

Recent advances in neural network modeling and theory, combined with advances in technology and computing power, are producing impressive results in a wide range of application domains. For example, large-scale deep-belief neural networks and convolutional networks now represent the state-of-the-art for speech recognition and image segmentation applications [Farabet et al., 2013; Mohamed et al., 2012]. However, the mostly sequential and synchronous clocked nature of conventional computing platforms is not optimally suited for the implementation of these types of massively parallel neural network architectures. For this reason a new generation of custom neuro-computing hardware systems started to emerge. These systems are typically composed of custom VLSI chips that either contain digital processing cores with dedicated memory structures and communication schemes optimized for spiking neural networks architectures [Furber et al., 2014; Neil and Liu, 2014; Wang et al., 2013], or full-custom digital circuit solutions that implement large arrays of spiking neurons with programmable synaptic connections [Merolla et al., 2014]. While these devices and systems have high potential for solving machine learning tasks and applied research problems, they do not emulate directly the dynamics of real neural systems. At the other end of the spectrum, neuromorphic engineering researchers have been developing hardware implementations of detailed neural models, using mixed signal analog-digital circuits to reproduce faithfully neural and synaptic dynamics, in a basic research effort to understand the principles of neural computation in physical hardware systems [Chicca et al., 2014; Douglas et al., 1995; Liu et al., 2002]. By studying the physics of computation of neural systems, and reproducing it through the physics of transistors biased in the subthreshold regime [Liu et al., 2002], neuromorphic engineering seeks to emulate biological neural computing systems efficiently, using the least amount of power and silicon real-estate possible. Examples of biophysically realistic neural electronic circuits built following this approach range from models of single neurons [Farquhar and Hasler, 2005; Hynna and Boahen, 2007; Mahowald and Douglas, 1991; van Schaik et al., 2010], to models of synaptic dynamics [Bartolozzi and Indiveri, 2007a; Liu, 2003; Xu et al., 2007], to auditory/visual sensory systems [Costas-Santos et al., 2007; Liu and Delbruck, 2010; Sarpeshkar et al., 1996; van Schaik and Meddis, 1999; Zaghloul and Boahen, 2004], to reconfigurable spiking neural network architectures with learning and plasticity [Chicca et al., 2014; Giulioni et al., 2008; Hsieh and Tang, 2012; Ramakrishnan et al., 2012; Yu et al., 2012]. In this work we propose to combine the basic research efforts with the applied research ones, by presenting a VLSI architecture that can be used to both carry out research experiments in computational neuroscience, and to

develop application solutions for practical tasks. The architecture proposed comprises electronic neuromorphic circuits that directly emulate the physics of real neurons and synapses to faithfully reproduce their adaptive and dynamic behavior, together with digital logic circuits that can set both the properties of the individual synapse and neuron elements as well as the topology of the neural network. In particular, this architecture has been developed to implement spike-based adaptation and plasticity mechanisms, and to carry out on-chip on-line learning for tasks that require the system to adapt to the changes in the environment it interacts with. Given these characteristics, including the ability to arbitrarily reconfigure the network topology also at run-time, we named this device the NP. The main novelty of the work proposed, compared to previous analogous approaches [Giulioni et al., 2008; Indiveri et al., 2006; Ramakrishnan et al., 2012; Yu et al., 2012] consists in the integration of analog bi-stable learning synapse circuits with asynchronous digital logic cells and in the embedding of these mixed-signal blocks in a large multi-neuron architecture. The combination of analog and digital circuits, with both analog and digital memory elements, within the same block provides the device with an important set of programmable features, including the ability to configure arbitrary network connectivity schemes. At the analog circuit design level, we present improvements in the neuron and spike-based learning synapses over previously proposed ones [Chicca et al., 2014; Indiveri et al., 2011], which extend their range of behaviors and significantly reduce device mismatch effects. At the system application level we demonstrate, for the first time, both computational neuroscience models of attractor networks and image classification neural networks implemented exclusively on custom mixed-signal analog-digital neuromorphic hardware, with no extra pre- or post-processing done in software. In the next section we describe the NP system-level block diagram, highlighting its dynamic and spike-based learning features. In Section 5.2 we describe in detail the circuits that are present in each building block. Finally, in Sections 5.8 we discuss the results obtained and summarize our contribution with concluding remarks.

5.2 Neuromorphic processor architecture

The block-diagram of the NP architecture is shown in Fig. 5.1. The device comprises a configurable array of synapse circuits that produce biologically realistic response properties and spiking neurons that can exhibit a wide range of realistic behaviors. Specifically, this device comprises a row of 256×1 silicon neuron circuits, an array of 256×256 learning synapse circuits for modeling long-term plasticity mechanisms, an array of 256×256 programmable synapses with short-term plasticity circuits, a 256×2 row of linear integrator filters denoted as "virtual synapses" for modeling excitatory and inhibitory synapses that have shared synaptic weights and time constants, and additional

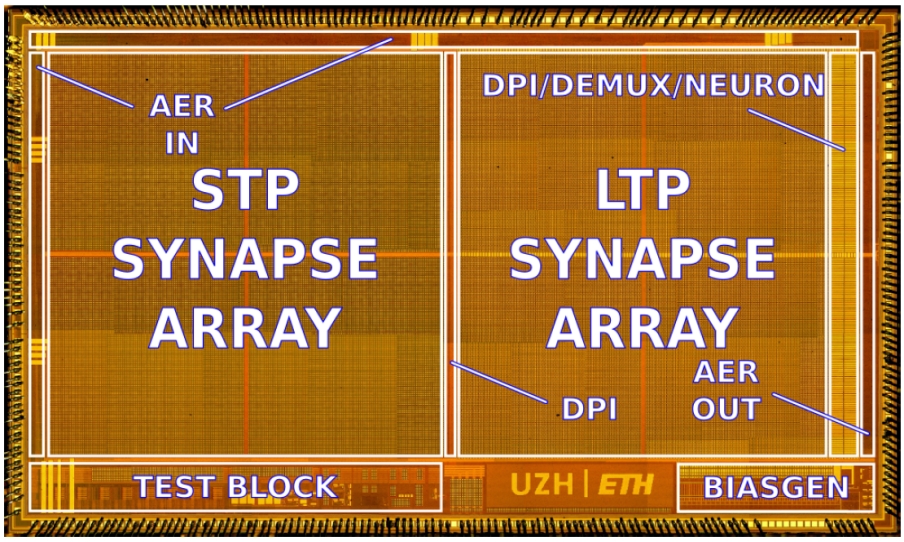


Figure 5.1: Micro-photograph of the NP. The chip was fabricated using a 180 nm CMOS process and occupies an area 51.4 mm^2, comprising 12.2 million transistors.

peripheral analog/digital Input/Output (I/O) circuits for both receiving and transmitting spikes in real-time off-chip. The NP was fabricated using a standard 180 nm CMOS 1P6M process. It occupies an areas of 51.4 mm^2 and has approximately 12.2 million transistors. The die photo of the chip is shown in Fig. 5.1. The area distribution of main circuit blocks is shown in Table 5.1. The silicon neurons contain circuits that implement a model of the adaptive exponential I&F neuron [Brette and Gerstner, 2005], post-synaptic learning circuits used to implement the spike-based weight-update/plasticity mechanism in the array of long-term plasticity synapses, and analog circuits that model homeostatic synaptic scaling mechanisms operating on very long time scales [Rovere et al., 2014]. The array of long-term plasticity synapses comprises pre-synaptic spike-based learning circuits with bi-stable synaptic weights, that can undergo either LTP or LTD, (see Section 3.3.2 for details). The array of Short-Term Plasticity (STP) synapses comprises synapses with programmable weights and STP circuits that reproduce short-term adaptation dynamics. Both arrays contain analog integrator circuits that implement faithful models of synaptic temporal dynamics (see Section 5.2.1). Digital configuration logic in each of the synapse and neuron circuits allows the user to program the properties of the synapses, the topology of the network, and the properties of the neurons. The architecture comprises also a "synapse de-multiplexer" static logic circuit, which allows the user to choose how many rows of plastic synapses should be connected to the neurons. It is a programmable switch-matrix that configures the connectivity between the synapse

Circuit	Dimensions ($\mu m \times \mu m$)	Number	Tot. Area: (mm^2)	(%)
Neuron	55.69×16.48	256	0.235	0.47
Post-synaptic learning	39.09×16.48	256	0.165	0.32
LTP Synapse	15.3×16.48	64 K	16.147	31.41
STP Synapse	16.24×16.48	64 K	17.129	33.32
Virtual synapse	35.6×16.48	512	0.300	0.58
Synapse de-mux	49.56×4389.4	1	0.218	0.42
AER in (columns)	8770×154	1	0.135	0.26
AER in (rows)	112×4357	1	0.488	0.95
AER out	166.2×4274.9	1	0.710	1.38
BiasGen	539.5×1973	1	1.064	2.07

Table 5.1: : Circuits area distribution. The remaining area used in the chip is occupied by the pads and additional test structures.

rows and the neuron columns. By default, each of the 256 rows of 1×512 synapses is connected to its corresponding neuron. By changing the circuit control bits it is possible to allocate multiple synapse rows to the neurons, thereby disconnecting and sacrificing the unused neurons. In the extreme case all 256×512 synapses are assigned to a single neuron, and the remaining 255 neurons remain unused. An on-chip programmable bias generator, optimized for subthrehold circuits [Delbruck et al., 2010a] is used to set all of the bias currents that control the parameters of the synapses and neurons (such as time constants, leak currents, *etc.*). An ADC circuit converts the subthreshold currents produced by selected synapse and neuron circuits into a stream of voltage pulses, using a linear pulse-frequency-modulation scheme, and transmits them off-chip as digital signals.

Finally, peripheral asynchronous I/O logic circuits are used for receiving input spikes and transmitting output ones, using the AER communication protocol [Boahen, 2000; Deiss et al., 1998].

5.2.1 Synapse temporal dynamics

In the NP all synapses process input spikes in real-time, as they arrive. Similarly the neurons transmit the spikes they produce immediately, as they are generated. In these types of architectures time represents itself and input data is processed instantaneously. There is no virtualization of time and no mechanism for storing partial results in memory banks. As a consequence, the circuits must operate with time-constants that are well-matched to those of the signals they are designed to process. Since this device is intended to be used in behaving systems that interact with the environment in natural real-world scenarios, it is important to design circuits that can implement a wide range of time constants, including very slow, biologically plausible, ones. To achieve this, and to model

neural dynamics with biologically plausible time constants, we used the DPI [Bartolozzi and Indiveri, 2007b]. This is a current-mode log-domain integrator. When biased in the subthreshold regime, this circuit can obtain long time constants, even with relatively small and compact capacitors. For example, in the 180 nm technology used, with a capacitor of 1 pF, we could obtain time constants of the order of tens of milliseconds without resorting to any advanced design techniques. However, to realize even longer time constants (e.g., of the order of hundreds of milliseconds), we used a shifted-source biasing technique, as described in [Linares-Barranco and Serrano-Gotarredona, 2003]. The synapse circuits in the two synapse arrays of the NP convert input voltage spikes into output currents which have non-linear dynamics, due to their adaptation or learning features. In addition, to model the synapse temporal dynamics, the currents produced by the circuit elements in the array are further integrated by a linear temporal filter. If we assume that all the synapses in an array have the same temporal dynamics (i.e., share the same time constants), then we can exploit Kirchhoff's current law and sum the output currents of all synapses in a row into a single DPI circuit. This allows us to save a significant amount of silicon real-estate, as we can use only one DPI per row, in each array. In particular, we use one excitatory DPI in the long-term plasticity array configured to produce time constants of the order of hundreds of milliseconds, to model the dynamics of N-Methyl-D-Aspartate (NMDA) receptors, and two DPI circuits (one for excitatory and one for inhibitory synaptic dynamics) in the short-term plasticity array, configured with time constants of the order of tens of milliseconds, to model the dynamics of AMPA and GABA receptors, respectively. We use the same principle for the 256×2 "virtual synapse" integrators in the architecture. These circuits comprise two DPI integrators per row (one for the excitatory synapse and one for the inhibitory one) with fixed sets of weights and shared time-constant parameters, biased to operate in their linear operating range. By time-multiplexing input spikes to a single virtual synapse we can model the effect of multiple independent inputs to the targeted neuron. For example, by stimulating the DPI with a single 10 KHz spike train, we can model the effect of 1000 synapses receiving a 10 Hz input spike train.

5.3 The silicon neuron block

The neuron circuit integrated in this chip is derived from the adaptive exponential I&F circuit proposed in [Indiveri et al., 2011], which can exhibit a wide range of neural behaviors, such as spike-frequency adaptation properties, refractory period mechanism and adjustable spiking threshold mechanism. The circuit schematic is shown in Fig. 3.1. It comprises an NMDA block ($M_{N1,N2}$), which implements the NMDA voltage gating function, a LEAK DPI circuit (M_{L1-L7}) which models the neuron's leak conductance,

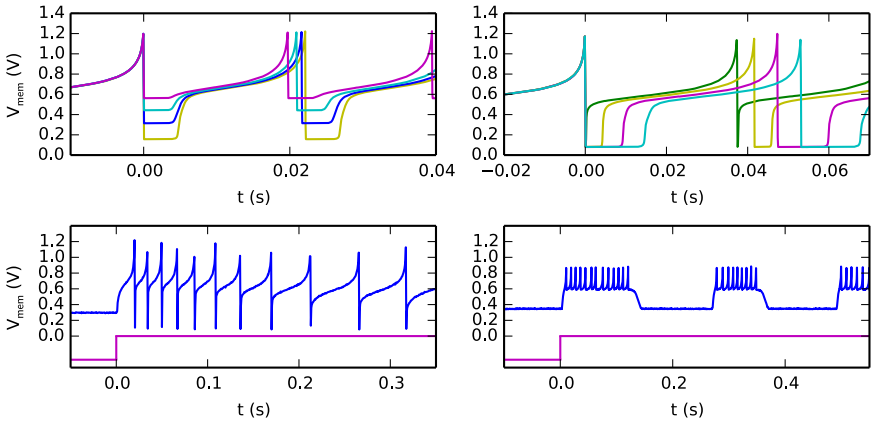


Figure 5.2: Different biologically plausible neuron's behaviors: (top-left) tunable reset potential, (top-right) tunable refractory period duration, (bottom-left) spike-frequency adaptation, (bottom-right) bursting.

an AHP DPI circuit (M_{A1-A7}) in negative feedback mode, which implements a spike-frequency adaptation behavior, an Na^+ positive feedback block ($M_{Na1-Na5}$) which models the effect of Sodium activation and inactivation channels for producing the spike, and a K^+ block (M_{K1-K7}) which models the effect of the Potassium conductance, resetting the neuron and implementing a refractory period mechanism. The negative feedback mechanism of the AHP block, and the tunable reset potential of the K^+ block introduce two extra variables in the dynamic equation of the neuron that can endow it with a wide variety of dynamical behaviors [Izhikevich, 2003]. As the neuron circuit equations are essentially the same of the adaptive I&F neuron model, we refer to the work of Brette and Gerstner [2005] for an extensive analysis of the repertoire of behaviors that this neuron model can reproduce, in comparison to, e.g., the Izhikevich neuron model. All voltage bias variables in Fig. 3.1 ending with an exclamation mark represent global tunable parameters which can be precisely set by the on chip Bias Generator (BG). There are a total of 13 tunable parameters, which provide the user with high flexibility for configuring all neurons to produce different sets of behaviors. In addition, by setting the appropriate bits of the relative latches in each neuron, it is possible to configure two different leak time constants (if_tau1! / if_tau2!) and refractory period settings (if_rfr1! / if_rfr2!). This gives the user the opportunity to model up to four different types/populations of neurons within the same chip, that have different leak conductances and/or refractory periods. An example of the possible behaviors that can be expressed by the silicon neuron are shown in Fig.5.2. The top-left quadrant shows measured data from the chip representing the neuron membrane potential in response to a constant current injection for different

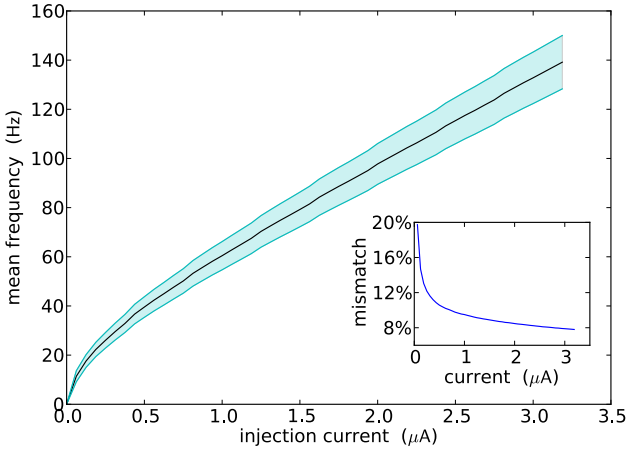


Figure 5.3: Population response of all neurons in the array to constant injection currents. The variance in the measurements is due to device mismatch effects in the analog circuits.

values of reset voltage. The top-right quadrant shows the neuron response to a constant current injection for different settings of its refractory period. The bottom-left quadrant demonstrates the spike-frequency adaptation behavior, obtained by appropriately tuning the relevant parameters in the AHP block of Fig. 3.1 and stimulating the neuron with a constant injection current. By further increasing the gain of the AHP negative feedback block the neuron can produce bursting behavior (see bottom-right quadrant of Fig.5.2). Figure 5.3 shows the F-I curve of all neurons in the NP (i.e., their firing rate as a function of the input injection current). The plot shows their average firing rate in solid line, and their standard deviation in the shaded area. The overall mismatch in the circuit, responsible for these deviations, is extremely small, if compared to other analog VLSI implementations of neural systems [Indiveri et al., 2006; Petrovici et al., 2014; Schmuker et al., 2014]. The average value obtained from the measurement results of Fig. 5.3 is only 9.4%. The reason for this improvement lies in the increased size of some critical transistors in the soma circuit – major contributor to neuron's mismatch. For example, the M_{L4} and M_{L5} FETs that set the neuron's leak time constants are of (W/L) size of ($2\,\mu$m/$4\,\mu$m) , while M_{Na3} and M_{Na4}, responsible for the firing threshold are of size ($4\,\mu$m/$0.4\,\mu$m) and ($1\,\mu$m/$4\,\mu$m) respectively. In addition to the neuron soma circuit, this block contains also post-synaptic plasticity circuits that are necessary for evaluating the weight update and "stop-learning" conditions described in Section 3.3.2. In particular these circuits integrate the spikes produced by the neuron into a current that models the neuron's Calcium concentration, and compare this current to three threshold currents

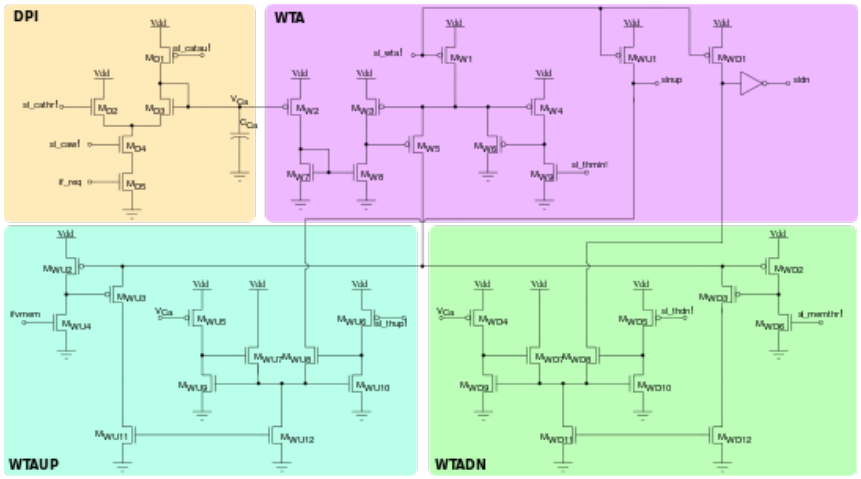


Figure 5.4: Post-synaptic learning circuits for evaluating the algorithm's weight update and "stop-learning" conditions. The DPI circuit M_{D1-5} integrates the post-synaptic neuron spikes and produces a current proportional to the neuron's Calcium concentration. Three current-mode winner-take-all circuits WTA, $WTAUP$, and $WTADN$ compare the Calcium concentration current to three set thresholds sl_thmin! , sl_thdn!, and sl_thup!, while the neuron's membrane current is compared to the threshold sl_memthr!.

that correspond to θ_1, θ_2, and θ_3 of eq. (3.10). In parallel, the neuron's membrane current (which is equivalent to the membrane potential in the theoretical model) is compared to an additional threshold equivalent to θ_{mem} of eq. (3.10). The schematic diagram of this circuit is shown in Fig. 5.4. The post-synaptic neuron's Calcium concentration is computed using the DPI M_{D1-D5}; the comparisons with the fixed thresholds are made using three current-mode WTA circuits M_{W1-W9}, $M_{WU1-WU12}$, and $M_{WD1-WD12}$. The digital outcomes of these comparisons set the signals slnup and sldn which are then buffered and transmitted in parallel to all synapses afferent to this neuron belonging to the long-term plasticity array.

5.4 The long-term plasticity synapse array

Each of the 256×256 synapse circuits in the long-term plasticity array comprises event-based programmable logic circuits for configuring both synapse and network properties, as well as analog/digital circuits for implementing the learning algorithm of Section 3.3.2. Figure 5.5 shows both digital and analog circuit blocks. The digital logic part, shown in Fig. 5.5a has an pulse generator circuit that manages the handshaking signals required by the AER protocol, and three one-bit configurable latches: one latch sets/resets the MON_EN signal, which enables/disables the synapse monitor circuit, which buffers the

synapse weight V_w signal for off-chip reading. The remaining two latches are used to set the BC_EN and REC_EN signals, which control the activation modes of the synapse. There are three different activation modes can be configured: direct activation, broadcast activation and recurrent activation. Figure 5.5b shows a timing diagram in which the relative latches for enabling broadcast and recurrent activation modes are configures in a synapse, using a 4-phase handshaking protocol. In the direct activation mode the synapse is stimulated by an AER event that has the matching row and column address. In the broadcast activation mode the synapse is stimulated by an AER broadcast event (that has a dedicated address word) which targets the matching column address. All synapses belonging to the same column that have the BC_EN bit set high get stimulated in parallel, when the matching broadcast event is received. In the recurrent activation mode the synapse of column j is stimulated when the on-chip post-synaptic neuron of row j spikes. Therefore it is possible to connect, internally, neuron i to neuron j by setting the REC_EN bit high of the synapse in row i and column j. In addition to these circuits, there is a pulse extender circuit which can increase the duration of the input pulse from nano-seconds to hundreds of micro-seconds. The schematic diagram of the analog/digital weight update circuits is shown in Fig. 5.5c. These circuits are subdivided into four sub-blocks: the SET block can be used to set/reset the bistable state of the synaptic weight by sending an AER event with the matching address and properly asserting the configuration signals set_hi and set_low. The JUMP block increases or decreases the synaptic weight internal variable (i.e., the voltage V_w) depending on the digital signals up and dn, that are buffered copies of the ones generated in the silicon neuron stop-learning block (see Section 5.3). The heights of the up and down jumps can be set by changing the delta_up! and delta_dn! signals. The BIST block consists of a wide-range transconductance amplifier configured in positive feedback mode, to constantly compare the V_w node with the threshold bi_thr!: if $V_w >$ bi_thr! then the amplifier slowly drives the V_w node, drifting towards the positive rail, otherwise it actively drives it toward the ground. The drift rates to the two states can be tuned by biases drift_up! and drift_dn! respectively. The current converter (CC) block converts the V_w voltage into a thresholded EPSC with maximum amplitude set by pa_wht!. Figure 5.6 shows experimental results that highlight the features of both synapse and neuron learning circuits in action: weight updates are triggered when the pre-synaptic spikes arrive, and when the post-synaptic neuron's Calcium concentration is in the appropriate range. Depending on the value of the Calcium concentration signal, the digital up and dn signal turn on or off. The weight internal variable is increased or decreased depending on where the membrane potential is with respect to the membrane threshold (see highlighted weight updates at t=273 and t=405). This variable is actively driven to the low or high bounds, depending if it is

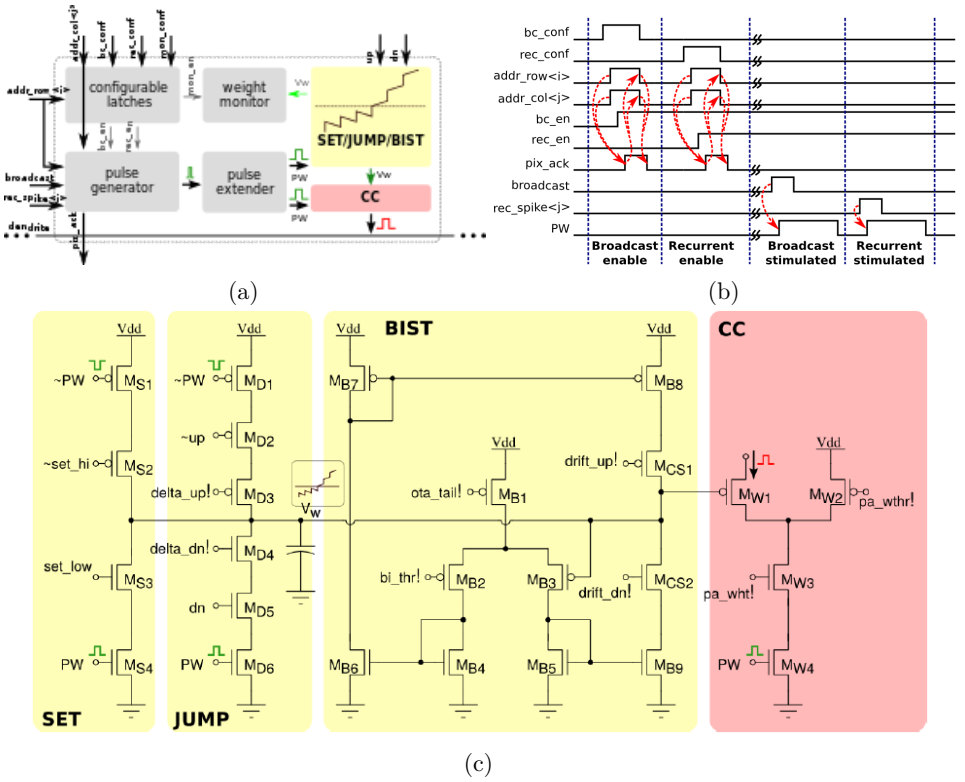


Figure 5.5: Long-term plasticity synapse array element. (a) Plastic synapse configuration logic block diagram. (b) Schematic diagram of the bi-stable weight update and current generator blocks.

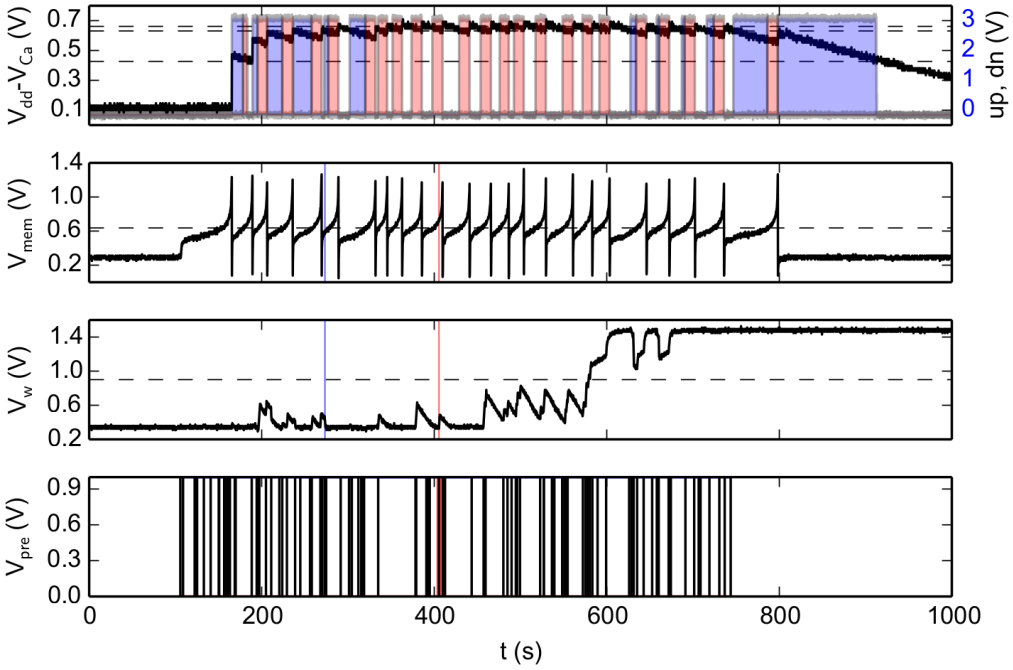


Figure 5.6: Spike-based learning circuit measurements. The bottom trace represents the pre-synaptic input spikes; the second trace from the bottom represents the bi-stable internal variable (node V_w of Fig. 5.5); the third trace represents the post-synaptic neuron's membrane potential; and the top trace shows both a voltage trace proportional to the neuron's integrated spiking activity as well as the digital control signals that determine whether to increase, decrease V_w, or leave it unchanged. The horizontal lines represent the thresholds used in the learning algorithm (see Section 3.3.2), while the vertical lines at t=273s and t=405s are visual guides to show where the membrane potential is, with respect to its threshold, for down and up jumps in V_w respectively.

below or above the weight threshold.

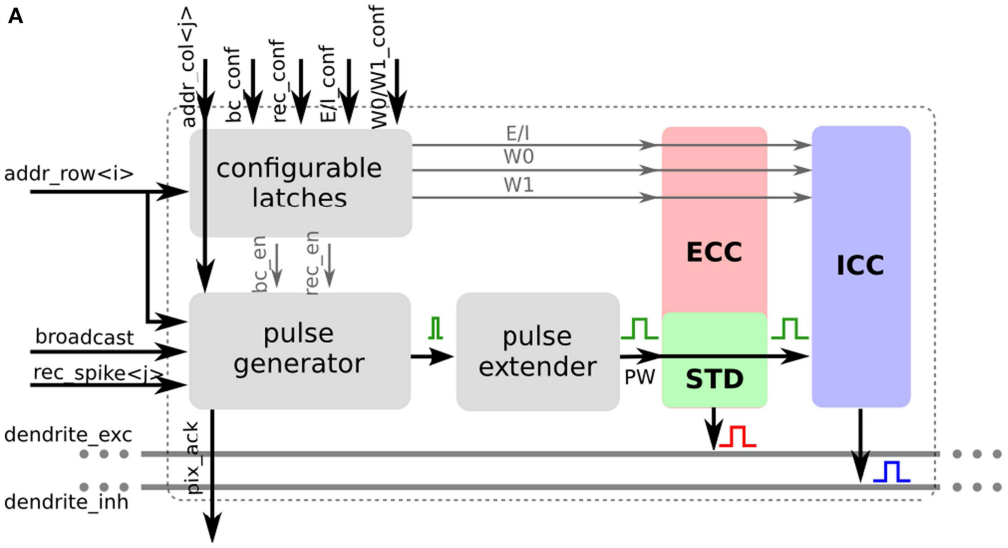
5.5 The short-term plasticity synaptic array

The array of short-term plasticity synapses contains circuits that allow users to program the synaptic weights, rather than changing them with a fixed on-chip learning algorithm. Specifically, each synapse has a two-bit programmable latch that can be used to set one of four possible weight values. In addition, it has an extra latch that can set the type of synapse (excitatory or inhibitory). In the excitatory mode, the synapse has additional circuits for modeling Short-Term Depression (STD) dynamics [Boegerhausen et al., 2003; Rasche and Hahnloser, 2001] whereby the magnitude of the EPSC decreases with every input spike, and recovers slowly in absence of inputs. Figure 5.7 shows both a block

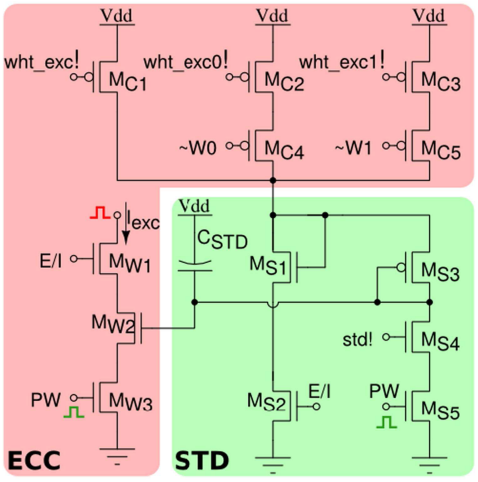
diagram of all synapse components, and the schematic diagram of the synapse analog circuits. In addition to the latches for setting the weight, there are two extra latches for configuring the synapse activation mode. As for the long-term-plasticity synapses, there are three possible activation modes: direct, broadcast, and recurrent (see Section 5.4). The left panel of Fig. 5.7b shows the excitatory current converter and the STD circuit. The current converter at the top generates a current that is proportional to the 2-bit weight. The proportionality constant is controlled through analog biases. This current charges up the C_{STD} capacitor through the diode connected p-FET M_{S3} so that at steady state, the gate voltages of M_{S1} and M_{W2} are equal. A presynaptic pulse on the PW port activates the I_{exc} current branch, and produces a current that initially is proportional to the 2-bit weight original current. At the same time, the PW pulse activates also the STD branch through transistor M_{S5} and an amount of positive charge that is controlled by the bias STD is removed from the capacitor C_{STD}. The gate voltage of M_{W2} is now momentarily lower than that of M_{S1}, and recovers slowly through the diode connected p-FET M_{S3}. Pulses that arrive before the capacitor voltage has recovered completely will generate a current that is smaller than the original one, and will further depress the effective synaptic weight through the STD branch. The excitatory block is only active if the E/I voltage is high. If E/I is low, the inhibitory current DAC in the right panel of Fig. 5.7b is active and generates a weight-proportional inhibitory current on PW pulses. Figure 5.8 illustrates how the STD behavior in the synapse: a spike burst was used to activate a programmable synapse. This resulted in a drop in synaptic efficacy during the later part of the burst. During a period of no stimulation the synapse recovered and responded with large Excitatory Post–Synaptic Potentials (EPSPs) to the initial part of the following burst, before depressing again. The responses to the two bursts are not identical in Fig. 5.8 as the state of the neuron, synapse, and DPI circuits are not exactly the same at the onset of each burst.

5.6 The peripheral input/output blocks

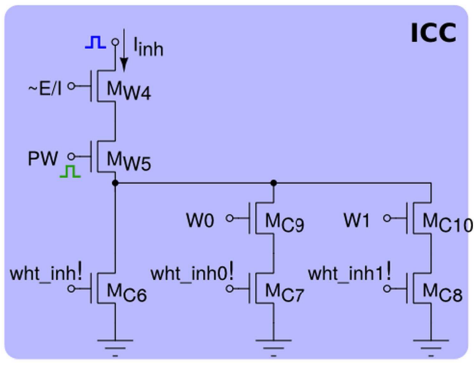
The peripheral digital circuits are used to transmit signals into and out of the chip. Given the real-time nature of our system, we use asynchronous digital circuits and quasi-delay-insensitive circuit design techniques [Manohar, 2006] to avoid discretization or virtualization of time. The AER communication protocol used encodes signals as the address of the destination synapse or as a control word for the input side, and as the address of the sender neuron in the output circuits.



(a)



(b)



(c)

Figure 5.7: Short-term plasticity synapse array element. (a) Block diagram of the synapse element. (b) Transistor level schematic diagram of the excitatory and inhibitory pulse-to-current converters.

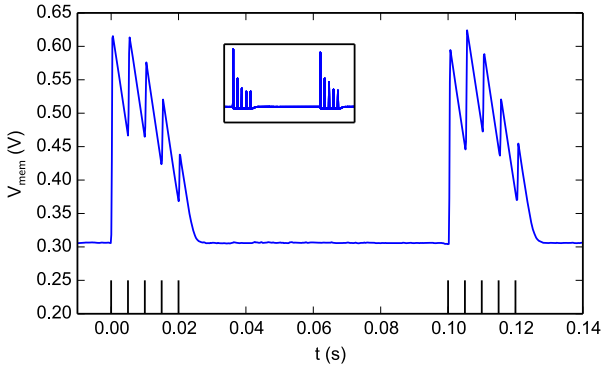


Figure 5.8: The effect of short term depression on EPSC magnitudes. Two bursts separated by 100ms were sent to a programmable synapse. Each burst has 5 spikes with an inter-spike interval of 5ms. Within a burst, The jumps in the neuron V_{mem} gradually get smaller as the synapse is depressed and the magnitude of the EPSCs it generates decreases. After the first burst, the synapse efficacy recovers as can be seen in the response to the second burst. The figure inset shows the derivative of the membrane potential which is equivalent to the synaptic EPSCs (minus the neuron leak).

5.6.1 The virtual synapse array

In this array there are two types of synapse circuits: one for representing the contribution of excitatory synapses, and one for the inhibitory synapses. Both excitatory and inhibitory synapses in this 256×2 array are designed to represent a dendritic branch that linearly sums their inputs coming from a large number of presynaptic sources, whose spikes are time multiplexed to the address line matching the synapse row address. For both excitatory and inhibitory synapses, inputs sent to them will have the same weight and the same dynamics. Both weight and synapse time-constant can be controlled through analog biases. The specific circuit that implements one of these virtual synapses is a DPI. The schematic of both excitatory and inhibitory virtual synapses is shown in Fig. 5.9. These synapses are not part of the recurrent connectivity matrix and can only be stimulated by external AER events.

5.6.2 AER Input/Output circuits

AER input circuits. Input spike events as well as chip configuration events are sent through a common input interface that uses a 21-bit address space. Input addresses are decoded into a total of 1249553 possible patterns subdivided into three categories: *Addressing*, *Local configuration* and *Global configuration*. *Addressing* inputs are decoded into a row and column address and are interpreted as a spike AE, which are sent to the

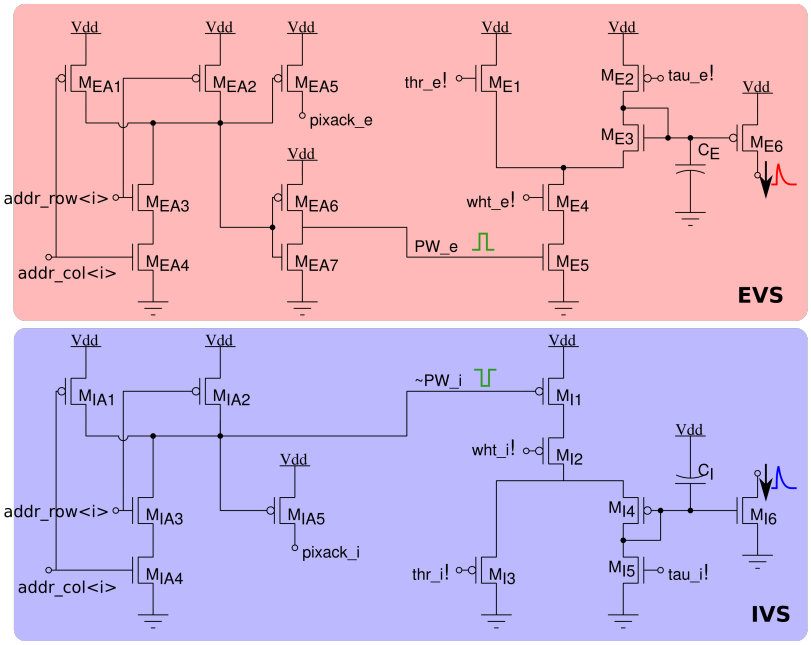
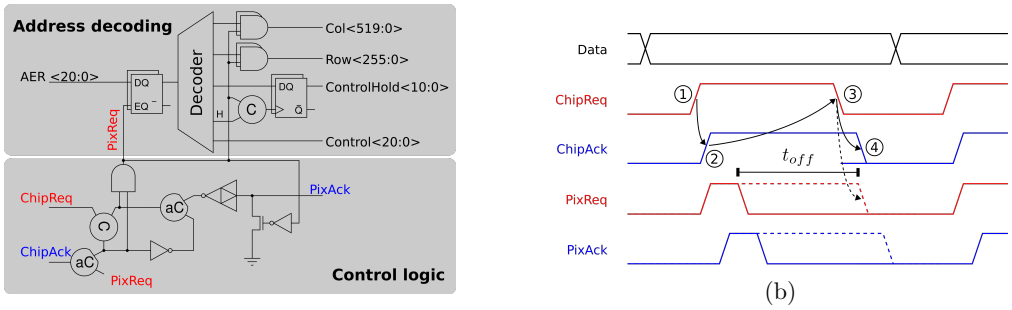


Figure 5.9: Top two panels: schematics of the virtual excitatory synapse and its associated DPI circuit. Bottom two panels: schematics of the virtual inhibitory synapse and its associated DPI circuit.



(a) (b)

Figure 5.10: (a) Input schematic comprising an address decoding block and a block for the control logic. (b) Four-phase handshaking protocol for sending inputs to the chip. The dashed lines show a common four-phase bundled-data protocol while the solid lines represent the proposed modification. The *pixel request* (PixReq) terminates before the *chip request* (ChipReq) and is thus not sensitive to off-chip delays (t_{off}).

desired target synapse of a target neuron. *Local configuration* address-events contain the row and column address of the target element as well as extra configuration bits that are written to the local latches of the addressed element. *Local configuration* patterns include commands for setting the type of synapse, programming its weight, or enabling broadcast or recurrent connections. Finally, the *Global configuration* inputs are decoded into configuration signals that represent global variables, stored onto registers in the periphery (rather than within the synapse or neuron elements). For example, the signals used to set the state of the synapse de-multiplexer are *Global configuration* signals. See the Supplementary material for additional details on these circuits. Figure 5.10a shows the schematic of the AER input circuits: the *Address decoding* block latches the input AER address bits upon a valid *pixel request* (see the PixReq signal in Fig. 5.10a) and decodes them into the appropriate input patterns. The *Control logic* block of Fig. 5.10a implements an extended version of a Mueller-Pipeline element [Sparsø, 2001] to decouple the internal *pixel request* termination from the external *chip request* (see the ChipReq signal in Fig. 5.10a and 5.10b) and therefore makes the length of the *pixel request* pulse independent from possible off-chip communication delays. This is necessary, as the duration of the *pixel request* directly affects the width of the pulse that produces the synaptic current. A full handshake transaction of the proposed input protocol is shown in Fig. 5.10b (solid lines) and compared to a standard 4-phase bundled-data protocol (dashed lines).

AER output. Each of the 256 neurons is assigned an 8-bit address for the output bus. When a neuron spikes, its address is instantaneously sent to the output AER circuits

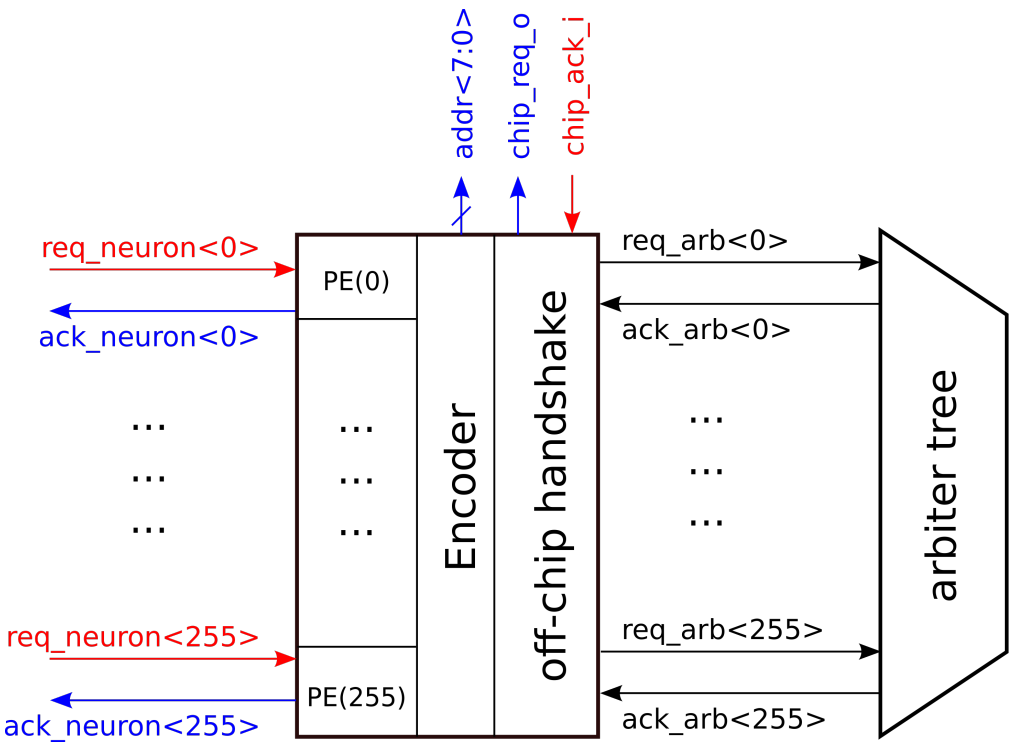


Figure 5.11: Block diagram of the output AER interface. See text for description.

using the common four-phase handshaking scheme. Although neurons operate in a fully parallel fashion, their address-events can only access the shared output bus in a serial fashion. To manage possible simultaneous spike collisions the output AER circuits include an arbiter circuit that only grants access to the external bus to one neuron at a time. Details of these circuits are provided in the Supplementary material. Figure. 5.11 shows a block diagram of the output AER interface. When a neuron spikes, its piplelining element immdeiately completes the 4-phase handshake with the neuron so that the neuron is reset and begins integrating input again. The stored request in the pipelining element is then forwarded to the arbiter. The arbiter selects one of the potential simultaneous requests from the pipelining elements, enqueing all other requests; when the arbiter circuits acknowledge a request from a neuron's pipelining element, the encoder block encodes the address of that neuron and puts it on the output address bus; at this stage, the off-chip handshake block initiates the AER four-phase handshaking mechanism with the external receiver. Once the external handshake completes, the arbiter resets the active pipelining element by deasserting its acknowledge line. A new request from one of the pipelining elements waiting in the queue can then be processed. While the arbitration

circuits are useful to preserve and transmit all spikes generated by the neurons (at the cost of introducing small delays in case of collisions), the pipelining circuits are useful for decoupling the neuron dynamics from the potential delays and the variability produced by the interaction with off-chip receivers. The fully asynchronous ouput AER interface was automatically generated, down to layout level, from behavioral hardware description language (HDL) code using the methodology presented in [Mostafa et al., 2013].

5.7 On-line learning and classification of visual stimuli in a neuromorphic feed-forward VLSI network

Neuromorphic systems are an ideal electronic substrate for real-time, low-latency machine vision [Delbruck and Lang, 2013; O'Connor et al., 2013; Serrano-Gotarredona et al., 2008]. Here we present a feasibility study which demonstrates how the NP can be used in conjunction with a spiking vision sensor for learning to solve an image recognition task. In this experiment we used a Dynamic Vision Sensor (DVS), interfaced to the NP via a commercially available digital board, used to route signals from the vision sensor to the chip. We implemented a two-layer spiking neural network which processes the visual stimuli by extracting sparse random features in real-time and projects them to 128 VLSI neurons on the NP. We trained the neurons to become selective to one of two image classes through a supervised learning protocol. The experimental protocol consists of showing a sequence of static images of objects from the Caltech 101 dataset coupled with a teacher signal to steer the activity of the output neurons. The DVS is put in front of a screen where the images are displayed. During the presentation, the images are presented with a small jitter around the center of the visual field to simulate microsaccadic eye movements. The movement causes the DVS retina to continuously stream spike trains corresponding to the edges of the objects in the image. The spike trains are then routed to the short-term plasticity synapse array stimulating a population of neurons corresponding to the hidden layer of the neural network. The spikes from the hidden layer neurons are internally routed to the plastic synapse array, thus activating the neurons of the output layer. A teacher signal is coupled to one of the presented classes and therefore it imposes a strong correlated activity in the corresponding subgroup of neurons in the output layer. To remove artifacts generated on the transition from one presentation of an image to the next, we gated the DVS spikes, simulating a saccadic suppression mechanism similar as, as observed in biology [Ross et al., 1996]. The performance of the experiment obviously depends on the right choice of parameters for the neural and synaptic dynamics. For this particular demonstration we chose to disable most of the complex aspects of the neural dynamics and optimized neuron and synapse parameters to obtain reasonable activity patterns in the hidden layer neurons. The activity in this layer is indeed the

most important since it drives the plasticity of the output layer synapses. After training, our classification system was able to respond selectively to natural images of cars and motorbikes taken from a dataset used in machine learning. Although an extensive characterization of the system's ability to perform object recognition is out of the scope of this work, we draw the following conclusions from our experiment: The choice of fixed, random projections from the input layer was surprisingly effective, though certainly not optimal for the task at hand. A better solution would be to include an unsupervised learning stage in the training protocol to optimize the weights of the convolution layer as in traditional machine learning approaches [Le et al., 2012; Le Cun et al., 1998] and in neural systems [Masquelier et al., 2009; Nessler et al., 2009; Olshausen and Field, 1997]. However, this stage typically requires the presentation of a large number of patterns, sophisticated synaptic plasticity rules, or offline learning [Merolla et al., 2014]. An other alternative could consist in interconnecting the neurons in the hidden layer to form a Liquid State Machine (LSM) [Maass et al., 2002]. This solution is particularly interesting in situations where information hidden in the fine temporal structure is expected to impact the performance of the recognition system. In our setup the stimuli have very poor temporal structure by design (the images presented are paired with random uncorrelated movements) and so it is not clear how the algorithm would benefit from the LSM. Similar to an LSM, our network of randomly connected neurons projects the input stimuli into a high-dimensional space where they can be classified by linear models but with far less parameter optimization [Barak et al., 2013]. This strategy is related to some of the state-of-the-art machine learning algorithms for pattern classifications, such as Super Vector Machines (SVMs) [Vapnik, 1995]. Clearly, the generalization properties of our system are not comparable to standard machine learning approaches but they are also expected to scale with the number of randomly connected neurons in the hidden layer [Barak et al., 2013; Rigotti et al., 2010]. Furthermore, given the properties of typical DVS output for object recognition in static scenes, a more appropriate choice of weights' spatial distribution seems appropriate, simply reflecting on an appropriate choice of connection matrices on the neuromorphic chip. This choice would certainly have a positive impact on classification performance. Notice also that we haven't exploited any temporal structure of the input data, though we recently demonstrated that our hardware supports this functionality [Sheik et al., 2012a,b, 2013]. These factors can be made even more relevant by appropriate choices of the parameters that we disabled in this experiment such as spike frequency adaptation. The use of multiple output neurons clustered into two distinct pools for our two-class discrimination problem, instead of simply two neuron units, is dictated by the imprecise (or "weak") nature of the classifiers. The rationale behind this choice is that given the many sources of noise in the system (the

micro-saccadic movements, the DVS spiking output, the stochastic plasticity mechanism, the hardware mismatch), each neuron taken singularly is not expected to perform well on the task (i.e., it will show low class specificity), while the performance improves if aggregated responses are considered. This can be visually appreciated from the raster plots of Fig. 5.13 where only population firing rates happen to be selective for the input classes but not the single neuron activities. This phenomenon is directly related to a notorious machine learning technique that uses "boosting" to improve the performance of weak-classifiers [Breiman, 2001; Schapire and Freund, 2012].

5.8 Discussion and Conclusions

Unlike conventional von Neumann type processors that carry out bit-precise processing, but access and store data in a physically separate memory block, the NP uses elements in which memory and computation are co-localized. The computing paradigm implemented by these types of neuromorphic processors does not allow for the virtualization of time, with the transfer of partial results back and forth between the computing units and physically separate memory banks at high speeds. Instead, their synapse and neuron circuits process input spikes on demand as they arrive, and produce their output responses in real-time. Consequently, the time constants of the synapses and neurons present in these devices need to be well matched to the signals the system is designed to process. For the case of real-time behaving systems that must interact with the environment, while processing natural signals in real-time, these time constants turn out to be compatible with the biologically plausible ones that we designed into the NP. As we implemented non-linear operations in each synapse (such as short term depression or long term plasticity), it is not possible to time-multiplex linear circuits to reduce the area occupied by the synaptic matrix array. As a consequence, our device is essentially a large memory chip with dedicated circuits for each synapse that act both as memory elements and computing ones. This approach is complementary to other recent ones that focus on accelerated neural simulations [Bruederle et al., 2011], or that target the real-time emulation of large populations of neurons, but with no on-chip learning or adaptive behaviors at the synapse level [Benjamin et al., 2014]. We presented a mixed-signal analog/digital VLSI device for implementing on-line learning spiking neural network architectures with biophysically realistic neuromorphic circuits as STP synapses, LTP synapses and low-power low-mismatch adaptive I&F silicon neurons. The proposed architecture exploits digital configuration latches in each synapse and neuron element to guarantee a highly flexible infrastructure for programming, with the same device, diverse spiking neural network architectures. All the operations of the chip are achieved via asynchronous AE streams. These operations include sending input spike events, configuring topology of

neural network, probing internal variables, as well as programming internal properties of synapse and neurons. The parameters for different synapse and neuron behaviors can be fine tuned by programming the temperature-compensated on-chip bias generator. The NP can be used to carry out basic research in computational neuroscience, and can be exploited for developing application solutions for practical tasks. In particular, this architecture has been developed to study spike-based adaptation and plasticity mechanisms, and to use it's ability to carry out on-chip on-line learning for solving tasks that require the system to adapt to the changes in its input signals and in the environment it interacts with. In Chapter 6 we present system level experimental results showcasing examples computational neuroscience models.

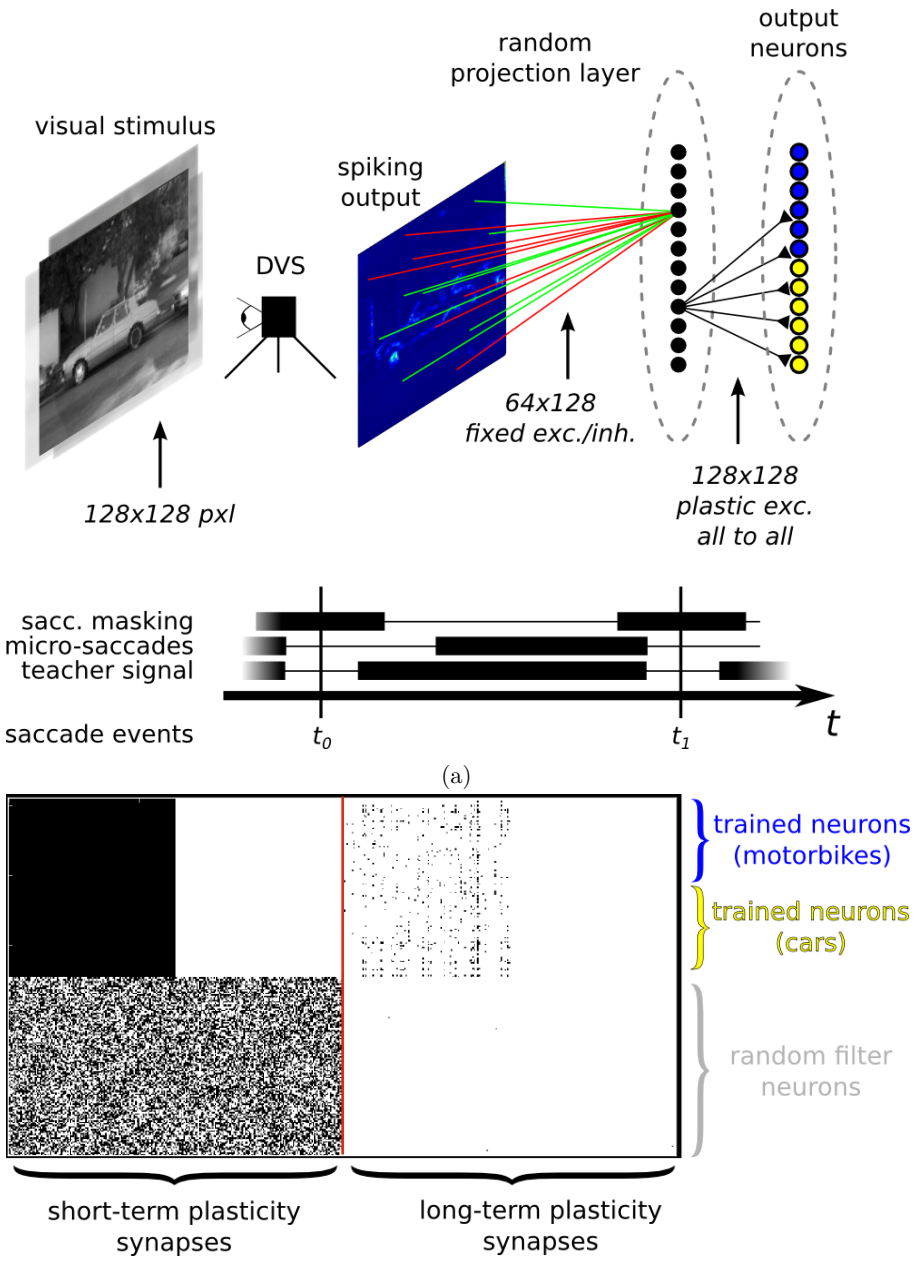


Figure 5.12: (a) Image classification using a spiking Dynamic Vision Sensor and the neuromorphic processor. Top: description of the neural network. Images grouped into two classes (here motorbikes and cars) are displayed on a screen with a small jitter applied at 10 Hz. The spikes emitted by the DVS are mapped to 128 hidden layer neurons, each receiving events from 64 pixels in random locations with positive or negative weights set at random. The output neurons receive spikes from all the hidden layer neurons through the plastic synapses array. Their activity is determined by a teacher signal correlated with one of the image classes. Bottom: diagram of the experimental protocol timeline. Notice the presence of a saccade inhibition mechanism which shuts down DVS input during a virtual saccade, i.e., when the displayed image is replaced with the next one. (b) Hardware implementation of the neural network. The short-term plastic synapses represent the synapses of the hidden layer.

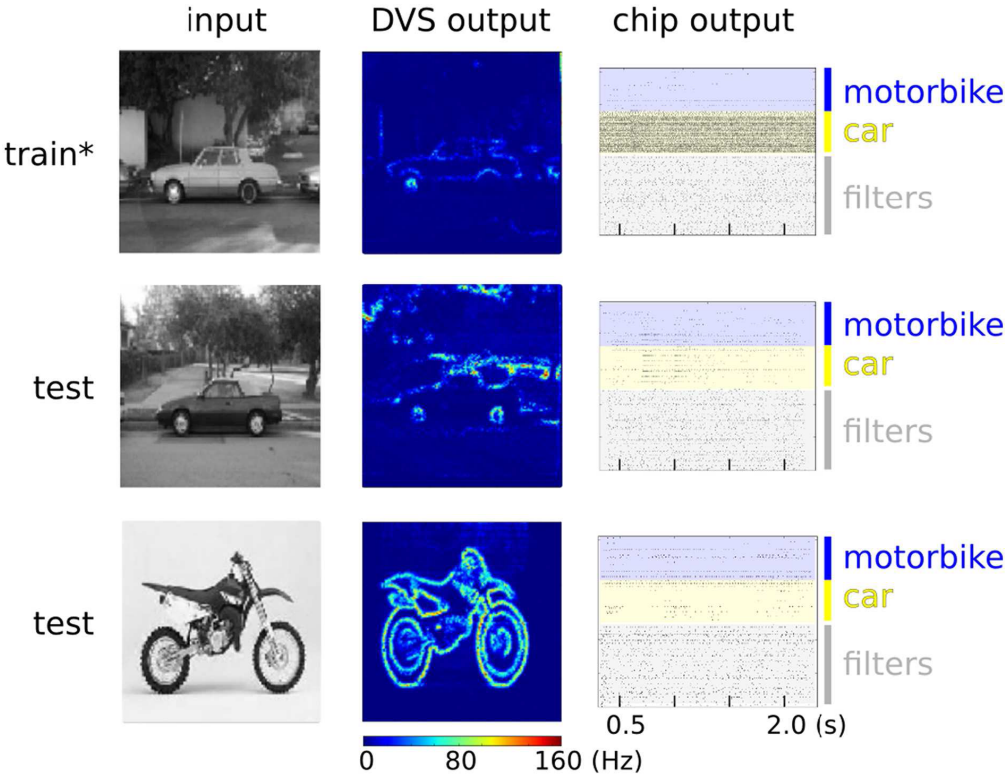


Figure 5.13: Spiking activity of the hardware neurons during training (upper panel) and testing phase (middle and lower panels). Left column: examples of raw images from the Caltech101 database. Middle column: heat map of the DVS spiking activity, where each pixel color represents the pixel's mean activity. Right column: raster plots of spiking activities. The star on the top panel label indicates that during the training phase an additional excitatory "teacher" signal stimulates the "car" output neurons to induce plasticity. During testing, only the excitatory currents from the input synapses drive the classifier activities. The average firing rates of the output layer for "motorbike" and "car" neuron pools are 6.0 Hz and 80.9 Hz during training. During testing with a "car" image they are 7.1 Hz and 11.1 Hz. During testing with a "motorbike" image they are 7.4 Hz and 4.9 Hz.

CHAPTER 6

Computing with attractor networks on neuromorphic VLSI systems

If we had access to billions of synthetic neurons that perfectly reproduce their biological counterpart, there would still be the problem of how to wire them to achieve interesting computation. In this chapter we explore some of the most advanced computational theories and we apply them in spiking neuromorphic hardware systems. A real device is a test-bench for neural models, to verify how robust they are in the face of problems not foreseen in the comfortable space of abstract mathematics. In addition, testing hardware devices should suggest new ideas and solutions to improve the models. Thus theory and hardware should be developed in parallel to benefit from each other. In this chapter we explore alternative signal processing and computational approaches for shaping new generations of computing technologies that implement brain-inspired models of neural computation. In particular we focus on how to map event-based computational models onto the neuromorphic processors presented in Chapters 4, and 5.

6.1 Attractor Networks as primitives of neural computation

The study of the collective dynamics of multiple neural populations with attractor states has been the subject of a good deal of investigation. This class of network is considered a basic building block for expressing different forms of computation in many different neural systems. In particular, reverberating states of cortical activity are thought to underlie important cognitive processes and functions. It has been shown for example that attractor networks in cerebral cortex are important for: long-term memory [Amit, 1992; Wittenberg et al., 2002], short-term memory [Amit and Brunel, 1997; Del Giudice et al.,

2003; Wang, 1999], contextual mental states [Rigotti et al., 2010], attention [Ganguli et al., 2008], bistable perception [Braun and Mattia, 2010; Gigante et al., 2009], and perceptual decision making [Wang, 2012, 2008]. In biologically inspired neural network models, it has often been assumed that an attractor in phase space represents an internal or an external source of information [Amari, 1977; Hopfield, 1982]. In other words, the information coming from the environment and the internal states is embodied in the network dynamics. In general terms, the ongoing dynamical activity in the cortex may be characterized by a multitude of attractor states. From a biological perspective, recurrent spiking neural network models have expressed the stable states of network dynamics in their firing rates. In this work we focus on attractor models that can be used as abstract models of populations of cells of the neocortex. This means that we will mostly deal with networks that have sparse connectivity levels and a local learning rule, and whose energy function can be shaped by input stimuli. The learning rule that we use is a local learning rule because it only exploits the spiking activities of pre and post-synaptic neurons to update the state of a single synapse, and therefore it enables efficient and parallel computing. Another important aspect of our learning rule is that it is not only local in space, but it is also local in time. This means that the training examples can be presented one at a time, and this gives an incremental learning rule. Algorithms that are based on these two concepts of locality in space and time are also considered to be biologically plausible.

6.1.1 The Energy function

An attractor network is a special class of recurrent network whose dynamics make the system collapse into a subspace of its available states. In such state neuronal cells are repeatedly active together. When an input is presented to the network, the network dynamics are steered towards a particular pattern of activity, and once the stimulus is removed, the evolution of the dynamics collapses into one of the available attractor states. The natural cognitive analogy is such that we can define a class of stimuli by the fact that all its members will cause the activation of one attractor state, represented by the activity of a cluster of neural cells. The area of the space in which the system moves towards the stable attractor state is called basin of attraction and its boundaries define one general class of inputs. Attractor networks have been formulated, first in an influential work by [Hopfield, 1982], as a physical magnetic system made of bipolar stochastic units. In this case an analogy can be made to the Ising-spin system that has been extensively used in physics to describe the interactions in ferromagnet and for which a large number of mathematical tools exist. The energy function for a network with binary-coded or continuously valued activity can be derived under the necessary condition of the existence

of a symmetric weight matrix. Under this hypothesis the energy functions are functions that describe a continuous decrease of the network activity, except at fixed points where both the value of the energy function and the activity of the network are constant. The energy function can be referred as the Hamiltonian, Lyapunov or objective function. The use of an energy function can be helpful in the analysis of attractor networks as it can be used to estimate the capacity and speed of convergence to an attractor state [Hertz et al., 1991]. The energy function can be formally expressed as an energy potential function $V(x)$, in which x represents the governing dynamics of the system.

$$\frac{dV(x)}{dt} \leq 0 \tag{6.1}$$

6.1.2 Effective Transfer Function for large recurrent networks

A mathematical description that naturally deals with the mismatch in neuromorphic systems is the mean-field approximation. This framework applies to large network of point like neurons. The input to a single neuron is the sum over all the synaptic contributions, that are considered to be distributed according to a Gaussian variable. In these models it is essential that the current dynamics of neurons is much faster than the depolarization dynamics. In other words we are in a diffusion approximation, therefore the synaptic efficacies are small and neurons receive a large number of synaptic contacts. In these conditions, we can express the mean synaptic input to a neuron, in an interval of its depolarization time constant τ as:

$$\langle I(t) \rangle = \tau \sum_j J_j \nu_j(t) + \langle I_{ext} \rangle - \beta_j \tag{6.2}$$

J_j is the synaptic efficacy defined as the jump of the membrane potential caused by a unitary spike, $\nu_j(t)$ represents the average spiking rate of neuron j in the network, I_{ext} is the external input current and β is a leak current term. Neurons in a network are organized in populations or pools, all neurons that are part of the same pool are statistically equivalent, and neurons in different pools have different parameters in terms of mean synaptic efficacy, mean constant leak, etc. The mean input current to a neuron in a pool α can be expressed as:

$$\langle I_\alpha(t) \rangle = \tau \sum_\gamma C_{\alpha\gamma} \langle J_\alpha \rangle_\gamma \nu_\gamma(t) + \langle I_{ext} \rangle - \beta_j \tag{6.3}$$

where γ is the total number of populations in the network. ν_γ is the mean rate of a neuron in population γ and $\langle J_\alpha \rangle_\gamma$ is the average synaptic efficacy of all synaptic contacts between populations γ and α. $C_{\alpha\gamma}$ is the average probability of connection between one

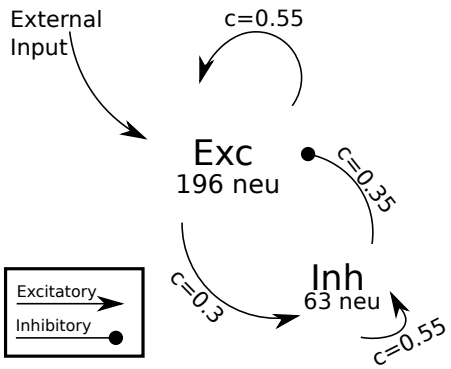


Figure 6.1: Realization of an attractor network architecture with two populations of neurons. The excitatory pool contains 196 neurons, while the inhibitory pool counts 63 neurons. Connections are random and sparse, C indicates mean connectivity in the network (see Section 6.1.3).

neuron in population α and all other neurons in population γ. In a large network, neurons are assumed to emit spikes following a Poisson process, and the state of a population of neuron is defined by the mean average firing rate of neurons in every pool. When the input current consists of a large number of independent contributions (i.e. $\tau C_\gamma \nu_\gamma \gg 1$ and $J_{eff} \ll \theta$), the central-limit theorem ensures that the sum over all Poisson independent distributions results in a Gaussian distribution. The input current is converted by a response function of a population of neuron as:

$$\nu_{out} = \Theta(\mu, \sigma) \tag{6.4}$$

This population transfer function only depends on the mean μ and variance σ of the total synaptic input current. The particular form of the transfer function depends on the neuron model used; in our case Θ also differs from population to population because of the different parameters such as spiking thresholds, mean average connectivity, leak rate, etc. Equation 6.1.2 expresses the mean spike rate of a population of neurons given the mean and variance of its input and we refer to it as the self-consistency equation. The mean and variance are in turn functions of neuronal activity in all the connected populations. This self-consistency equation evidences stable points in the network dynamics when the input and output are equal, that is along the green bisect line in figures 6.2a 6.2b. We can also write a system of self-consistency equations; such equations express the condition that all neurons in every sub-population produce an output that is compatible with their inputs. In such a description we can consider the entire network as a local cortical module composed of sub-populations and write a system of equations in this

form:

$$\begin{cases} \nu_1 = \Theta_1(\mu_1(\vec{\nu}), \sigma_1(\vec{\nu})) \\ \nu_2 = \Theta_2(\mu_2(\vec{\nu}), \sigma_2(\vec{\nu})) \\ ... \\ \nu_n = \Theta_n(\mu_n(\vec{\nu}), \sigma_n(\vec{\nu})) \end{cases} \tag{6.5}$$

where $\vec{\nu} = (\nu_1, \nu_2, .., \nu n)$ is a N-dimensional vector of the mean rates of neurons in each of the sub-populations. The solutions to the system are stationary states of network dynamics; in these points the energy function is constant. The stable subset of these points are attractors of the dynamics (green points in figures 6.2a 6.2b). Experimentally one could focus on a single population a time, and consider the mean rate ν_1 of population 1 as parameter. One then analyses the stable points of the remaining populations induced by the rate ν_1 of the population under focus and by the full feedback among all other populations. This means freezing one population of neurons to fire at a rate ν_1 and letting all other populations in the pool react to that input. Formally [Mascaro and Amit, 1999] solved the system of N-1 equations:

$$\begin{cases} \nu_2 = \Theta_2(\mu_2(\bar{\nu_1}, \nu_2, .., \nu_n), \sigma_2(\bar{\nu_1}, \nu_2, .., \nu_n) \\ ... \\ \nu_n = \Theta_n(\mu_n(\bar{\nu_1}, \nu_2, .., \nu_n), \sigma_n(\bar{\nu_1}, \nu_2, .., \nu_n) \end{cases} \tag{6.6}$$

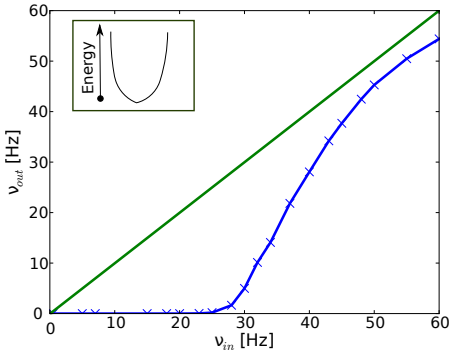
in which $\bar{\nu_1}$ is fixed. The inputs of the population under examination will drive the neurons to a new rate $\vec{\nu}^{'}(\bar{\nu_1})$ that is in general different from $\bar{\nu_1}$. This new rate is expressed as:

$$\vec{\nu}_{1out} = \Theta_1 \left(\mu_1(\bar{\nu_1}, \vec{\nu}^{'}(\bar{\nu_1}), \sigma(\bar{\nu_1}, \vec{\nu}^{'}(\bar{\nu_1})) \right) = \Theta_{eff}(\bar{\nu_1}) \tag{6.7}$$

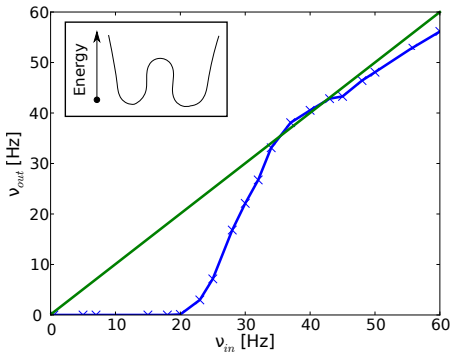
This describes the consequence of exciting a population at a fixed rate $\bar{\nu_1}$. By sweeping along a range of input frequencies we can obtain a measure of the function $\Theta_{eff}(\nu)$ which describes the full feedback response of the entire system. If the state of populations varies continuously, also the Effective Transfer Function (ETF) varies in a continuous way. It is however possible that for some ν_1 values the state of the populations can undergo a discontinuity; this would happen when the state trajectory become unstable in some point v_*.

6.1.3 Spike based simulation of an attractor network

To demonstrate that it is possible to introduce stable attractor states in a spiking neural network by only varying recurrent excitatory weights, I performed a spiking simulation of a network composed of 259 neurons. This particular case is interesting because synaptic



(a) Mono-stability, Effective Transfer Function



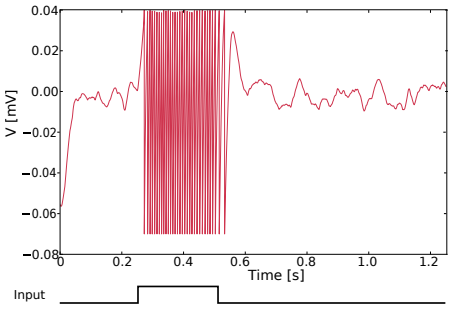
(b) Bi-stability, Effective Transfer Function

Figure 6.2: Gain regulation and stability of attractor states. The ETF is represented by the blue continuous line. The green line is the bisect. Green dots represent stable states of the network dynamics, i.e. minima in the energy landscape. Red dot is an unstable state of the dynamics. By varying the mean recurrent synaptic efficacy of the excitatory sub-population of neurons the network exhibits mono-stability 6.2a or bi-stability 6.2b.

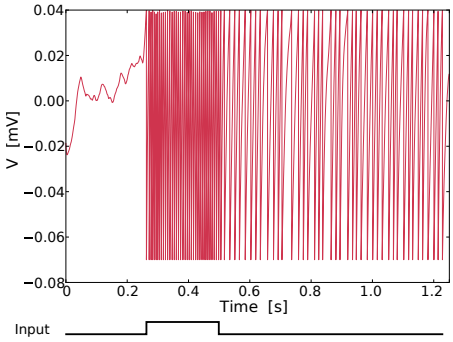
plasticity rules can induce weight changes and therefore introduce stable states in the network dynamics, in such scenarios stable states represent memories. In particular, the network architecture visible in Fig. 6.1 has been simulated for different values of mean recurrent excitatory weight. The network is composed of two populations of neurons, the excitatory population is composed of 196 excitatory neurons while the inhibitory population is composed of 63 neurons. The synaptic connectivity values in the network are initialized at random and they are sparse. Connectivity levels are indicated in Fig. 6.1. An external input is used to drive all the neurons in the pool of excitatory neurons via full connectivity. Neurons are instantiated with an exponential integrate-and-fire neuron model and their membrane potential V is described by the following equation:

$$\frac{dV}{dt} = \frac{1}{\tau_m}(-V + V_{res} + g_e + g_i + \Phi) + I_{ext} \tag{6.8}$$

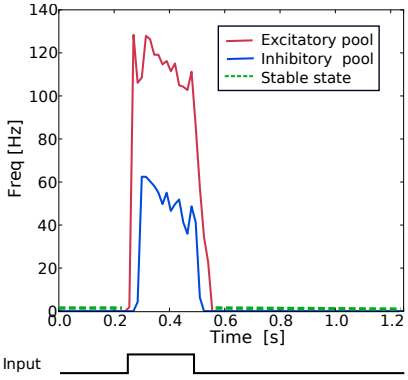
in which V_{res} represents neurons' reset potential, τ_m is the integration time constant, Φ is a constant leak term, I_{ext} is an external source of current (used to simulate a Poisson noise contribution), and g_e, g_i represent the synaptic current contributions of excitatory



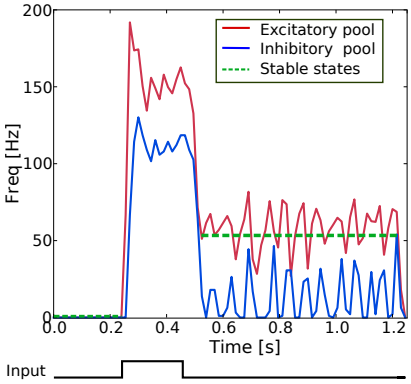
(a) Mono-stability.



(b) Bi-stability.



(c) Mono-stability, mean firing rate for the two pools of neurons.



(d) Bi-stability, mean firing rate for the two pools of neurons.

Figure 6.3: Computer simulations of a recurrent network. Membrane potential of a sample neuron in the excitatory population. The network exhibits mono-stability 6.3a or bi-stability 6.3b, depending on the mean recurrent excitatory weight. The square line at the bottom of every plot evidences the input stimulation protocol. The input to the network is a Poisson spike train of mean rate 40 Hz directed at the input of the excitatory pool of neurons via full connectivity. Note that the stable state in 6.3d is found around 50 Hz as predicted by the ETF in Fig. 6.2b.

and inhibitory synapses. These synaptic contributions obey the following dynamics:

$$\begin{aligned} \frac{dg_e}{dt} &= -\frac{g_e}{\tau_e}\sum_n \delta(t - t_{in}), \\ \frac{dg_i}{dt} &= -\frac{g_i}{\tau_i}\sum_n \delta(t - t_{in}), \end{aligned} \tag{6.9}$$

in which τ_e, τ_i are the excitatory and inhibitory time constants. The arrival of a spike a time t is represented with the δ function. The parameters used to simulate the network [1] are shown in table 6.1.3. Two sets of parameters have been selected to simulate mono- and bi-stable dynamics. In order to navigate the space of possible parameters the ETF has been measured. Figures 6.2a, 6.2b show the ETF for the excitatory pool of neurons for different mean recurrent excitatory synaptic efficacies. For low value of the mean recurrent excitatory efficacy (i.e. 1.9 mV), there exists only a single stable state in the network dynamics (i.e. the spontaneous activity state at low frequency in Fig. 6.2a). This is also evident in Figures 6.3a, 6.3c. In this simulation an external input is provided for a brief period of time ($0.22 < t < 0.45$). This stimulus drives the excitatory pool of neurons to fire at high rate (i.e. 120 Hz as in Fig. 6.3c). Once the input is removed the network jumps back to the spontaneous activity state. In this regime, neurons in the network only exhibit sub-threshold oscillations caused by the I_{ext} current contribution (see Fig. 6.3a). When the recurrent excitation is increased,the network dynamics exhibit a second stable state at high firing rate (predicted to be at 50 Hz by the ETF in Fig. 6.2b). The second stable point is observed when exciting the excitatory pool of neurons with a 40 Hz input. Once the stimulus is removed the network relaxes in the reverberant activity state, see Fig. 6.3d, and Fig. 6.3b.

Parameter name	**Value (mono-stability)**	**Value (bi-stability)**
τ_m	21 ms	21 ms
τ_e	8 ms	8 ms
τ_i	8 ms	8 ms
V_{thr}	40 mV	40 mV
V_{res}	-70 mV	-70 mV
we	**1.9 mV**	**2.18 mV**
wi	3 mV	3 mV
Φ	-5 mV	-5 mV

This simulation has shown that it is possible to introduce stable states in the network dynamics by only varying recurrent excitation in a network of spiking neurons.

[1] I used Brian Simulator [Goodman and Brette, 2008]

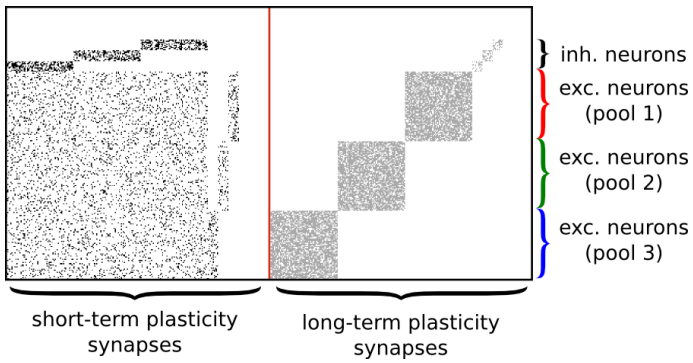
6.2 Attractor Networks in neuromorphic hardware

In this experiment we explored the collective dynamics of multiple populations of spiking silicon neurons that emulate the biophysics of cortical neurons organized in attractor networks [Amit, 1992]. In this work we configured a neuromorphic VLSI chip comprising spiking neurons and dynamic synapses to implement recurrent neural networks with excitatory and inhibitory connections (implementing positive and negative feedback loops respectively). Circuit and system level descriptions of the device have been presented in Chapter 5. Here, we configured the circuits to implement cortical neural network models and analyzed their dynamics by measuring the neuron's spikes and calculating their mean firing rates. We implemented the hardware attractor networks following the theories and methods proposed in [Amit, 1992; Amit and Mongillo, 2003; Del Giudice et al., 2003; Giulioni et al., 2012; Wang, 1999], and thanks to the measure of the ETF. We constructed an architecture comprising six pools of neurons recurrently connected. Specifically, there are three pools of 64 excitatory neurons and three pools of 10 inhibitory neurons. Every pool receives local excitation via recurrent connections implemented via the on-chip long-term synaptic plasticity circuits (see Fig. 6.4). The crossbar structure of the NP's synaptic matrix is such that horizontal lines represent dendrites of neurons while vertical lines represent axons. In Fig. 6.4 the dense dotted points represent synaptic contacts. The synaptic matrix is divided in two regions: the short-term plasticity synapses (left portion) and the long-term plastic synapses (right portion). The recurrent connectivity of the long-term plasticity synaptic contacts for the excitatory and inhibitory populations has connectivity parameters $c^e_{ee} = 0.7, c^e_{ii} = 0.4$ respectively, where c denotes the probability of connections between neurons (see dots in Fig. 6.4a). We configured the connectivity matrix of the short-term programmable synapses such that every excitatory pools of neurons is homogeneously connected with all other excitatory pools with excitatory connectivity parameter $c^e_{ee} = 0.2$ and inhibitory connectivity parameter $c^i_{ee} = 0.2$. Inhibitory pools of neurons are connected to their corresponding excitatory pools (e.g., inhibitory pool #1 is connected to excitatory pool #1) via inhibitory synapses, with a connectivity parameter $c^i_{ei} = 0.4$. Excitatory pools of neurons are connected to their respective inhibitory pools of neurons via the short-term plasticity excitatory synapses, with a connectivity parameter $c^e_{ie} = 0.7$. The behavior of the network when stimulated by an external transient stimulus is shown in Fig. 6.4b. The profile of the external stimulus is depicted by the square waves below the mean rate plot. Different colors indicate input for their respective population. Input stimuli are Poisson spike trains generated on a computer and sent via the AER to the on-chip virtual synapses. The mean rate of the input spike trains is $v_{in} = 100\,Hz$ and their duration is $t = 0.5\,s$. The results show the activity of the neural populations. When these

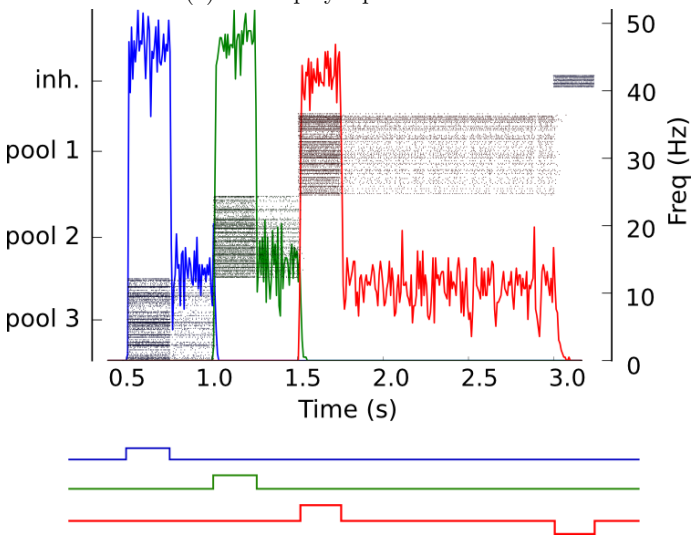
populations are being driven by external stimuli their activity reaches a mean rate of 50 Hz. After the removal of these stimuli the pools of neurons relax to an elevated state of activity at about 15 Hz, indicating that the neurons settled into their attractor states. This persistent activity is the neural correlate of working memory and can be exploited as an asynchronous distributed memory state that has peculiar dynamical properties of error correction, pattern completion and stability against distractors. If a population is in an attractor state, a transient stimulus to a different pool of neurons shuts down its activity via direct inhibitory connections (on the short-term plasticity synaptic matrix), and brings the newly stimulated pool of neurons into a new attractor state. If we inhibit an active pool of neurons with an external stimulus (e.g., at $t = 3\,\mathrm{s}$, with a Poisson stimulus of mean rate $\nu = 200\,\mathrm{Hz}$ via inhibitory synapses) the population is reset and becomes inactive. This experiment demonstrates how it is possible to implement real-time state dependent computation and reliable memory storage using sets of 64 slow and imprecise silicon neurons. A similar, but more elaborate experiment showing how these types of circuits can be used to synthesize context-dependent behavior in neuromorphic agents, in the context of cognitive computation is presented in [Neftci et al., 2013]. The implementation of plausible neural collective dynamics in neuromorphic substrates is an important step also for future nano-technologies that are likely to rely on mismatched and unreliable components.

6.3 Learning associative memories in neuromorphic hardware

In this paragraph I demonstrate robust formation of memories in a neuromorphic hardware network. In particular, I demonstrate that it is possible to structure an initial homogeneous network thanks to the use of input stimuli and without a teacher signal. The input signals induce synaptic changes in a fraction of synapses that are recurrently connected in an initial homogeneous and random fashion. These synaptic changes give rise to stable attractor states capable of sustaining self-persistent activity. The final structure of the network is described by clustered sub-networks that represent memories. The network is configured as a recurrent competitive network: I connected the chip's 256 neuron outputs to 90% of the 256 plastic LTP synapses, via recurrent excitatory connections, and to a random subset of 50% non-plastic STD inhibitory synapses, via recurrent inhibitory connections. The initial configuration of the excitatory recurrent connections is shown in the left box of Fig. 6.5a, where every white dot represents a synapse in state 1 (high) and every black dot represents a synapse in state 0 (low). Initially only 10% of the total synaptic contacts are randomly assigned to state 1 (see right box of Fig. 6.5a). I configured the parameters of the learning circuits to induce LTP and LTD transitions. These transitions have been calibrated to take place in an intermediate



(a) On-chip synaptic matrix



(b) Network behavior and stimulation protocol

Figure 6.4: Attractor network in neuromophic hardware. Three clustered pools of 64 neurons each are configured in an attractor network topology. (a) The on-chip synaptic matrix in which every dot represents an on-chip recurrent connection. (b) The output spiking activity of the network. Every pool of neurons is a competing working memory. The timing of the inputs is shown with square waves under the plot, the down jump of the red line represents the stimulus to the inhibitory pool of neuron responsible for switching off the on-chip memory.

range of firing rates (i.e., between 50 Hz and 150 Hz), and to stop changing the weights for higher frequencies. I then stimulated four distinct groups of neurons repeatedly, by sending spikes to the non-plastic synapses in a Poisson distributed manner, for one second each (see Fig. 6.5b). The square waves below Fig. 6.5b represent the presence of the input spike trains. As pattern are presented, the synaptic matrix changes structure in a way that reflect the input signal statistics, and accordingly to the binary weight update rule described in Section 5.2.1. At each stimulus presentation, the plastic synapses of neurons receiving both feed-forward input from the Poisson input spike trains and recurrent feedback from the output neurons tend to potentiate. The opposite happens for the plastic synapses of neurons that did not receive feedback spikes correlated with feed-forward inputs (see second, third and fourth plot of Fig. 6.5a). As it can be seen from the plot (Fig. 6.5a), the synapses are gradually clustering in four different regions of the synaptic matrix. By increasing the number of potentiated synapses in the network, the populations of recurrently connected neurons start to exhibit meta-stable states of the network dynamics. These states are sustained activity states, with higher and persistent firing rates. Also the peculiar pattern of activity reflects the structure of the inputs (see second, third and fourth plot of Fig. 6.5b). The attractor states are stable when enough recurrent connections potentiate, and the spiking activity of the population remains sustained even after the removal of the input stimulus (i.e. see activity during t=1-2s s, t=3-4 s, t=5-6 s, and t=7-9 s in Fig. 6.5c). The network structure implements competitive dynamics via fixed inhibitory recurrent connections. This is evidenced by the fact that when neurons belonging to different attractors are stimulated, their activity suppresses the activity of other attractors.

Figure 6.5c shows a neuron membrane potential measured at the beginning of the experiment, when there are no attractors; in the middle, as the attractors are being formed; and at the end of the experiment, during sustained activity. An important aspect of this learning process is that stable memories emerge in the device despite the high degree of variability (with a coefficient of variation of about 10% [Qiao et al., 2015]). In addition the learning process is always on, and there is no distinction between a learning phase and a testing phase. We exploited the fact that the input stimulus is driving the population of neurons at high firing rates, where the probabilities of LTP and LTD are maximized. In the frequency ranges of stable attractor states (i.e., about 20 Hz where there is persistent activity) LTP and LTD probabilities are almost equal to zero. This means that the network is learning when the stimulus is present, while it is not updating weights when the input stimulus is removed, as frequencies are much lower.

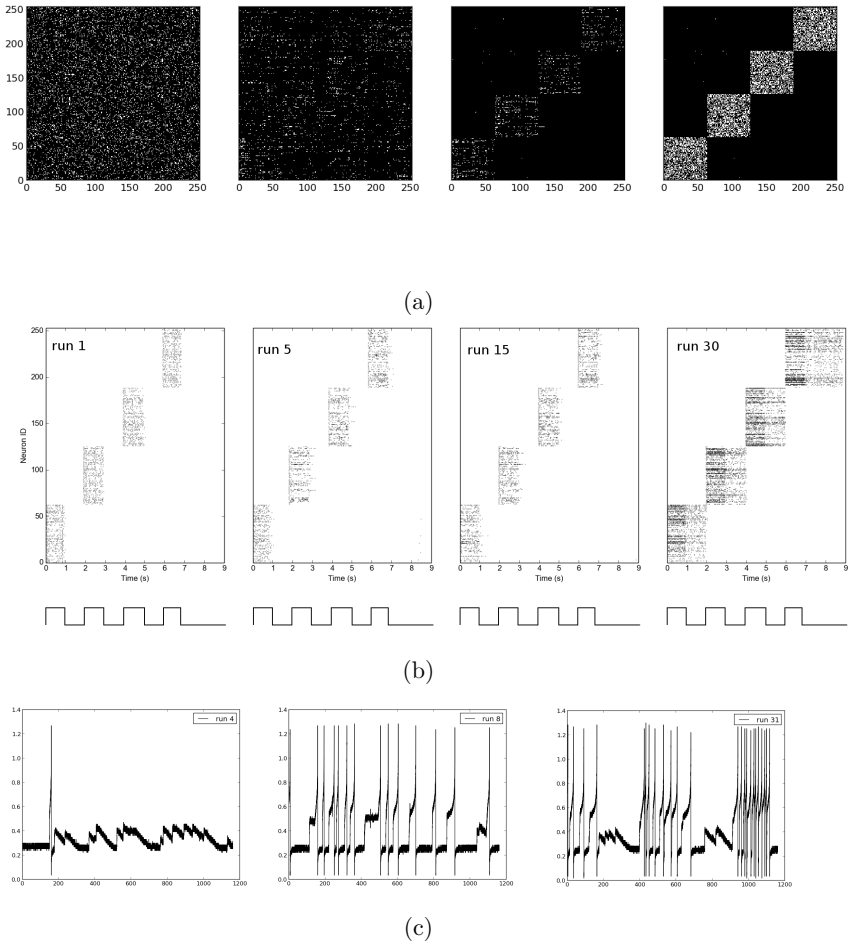


Figure 6.5: Forming stable attractor states in the neuromorphic processor. a) Evolution of the synaptic matrix during the learning experiment: white dots represent synapses in state 1 (high) and black dots represent synapses in state 0 (low). b) Evolution of network activity during learning, at the beginning there is no reverberant activity, while at the end there is formation of four stable attractor states. c) Membrane voltage of one neuron during stimulation at different stage of learning.

6.4 Information storage capacity of physical recurrent networks

The problem of estimating how many attractor-like memories can fit in a device is still an open question from a theoretical perspective. However, similar questions have been answered; they formalize the capacity of perceptron [Rosenblatt, 1958] with unbounded synaptic weight or the case of Hopfield network models[Hopfield, 1982]. Although our neuromorphic processor shares many similarities with Hopfield networks, only theoretically and with hard bounded synapses there exists a solution [Fusi, 2002]. Information in the Hopfield model is stored in stable attractor states. A stable state is a fixed point of the network dynamics. In this state each of the randomly connected neurons is computing the weighted sum of all its synaptic inputs, and does not update its state. All neurons are stalled at the same state and will not change from it. The information storage capacity can be understood also by looking at the examples of pattern completion and error correction. Storing a memory in a network means that there exists an attractor state. If we consider a network in which two patterns (attractors) have been stored, we can ask how well we can retrieve them. Thanks to the attractor dynamics we can retrieve a full stored pattern by providing only part of it. The network will then converge to the closest attractor state, resulting in the retrieval of a prototypical memory. Therefore the model is capable of error correction and pattern completion. It is interesting to know how many patterns K we can memorize in a network composed of N neurons. The question can be formally stated as "What is the maximum number of patterns K of N binary entries that can be made stable in a network of N neurons, and what is the synaptic structure most suitable for it? " Hopfield predicted with numerical simulations that $K \approx 0.15N$[Hopfield, 1982]. In a work by [Amit et al., 1985] it is shown the existence of a critical capacity $p = \alpha C$. They used a fully connected network composed of N neurons having C synapses on each dendrite, therefore the total number of synapses is NC. The constant $\alpha \approx 0.14$ was found numerically. Strikingly once this number of memories is surpassed, old memories are not forgotten while storing new ones, but all memories in the system become irretrievable. This represents an abrupt variation of regime in which none of the memories can be retrieved any more and it is named *catastrophic forgetting*. In hardware it should make sense to ask similar questions that relate the signal to noise ratio of stored memory patterns. In [Fusi and Abbott, 2007], the limits on the memory storage capacity of bounded synapses have been considered. In this work they analysed network of neurons interconnected by bounded synapses and they studied the number of stored memories together with the average memory lifetimes. They focused on the problem of overwriting synaptic modifications representing older memories while storing new ones. They refer to ongoing plasticity as a combination of two effects: first, synaptic modifications can occur caused by spontaneous activity; second, synapses that retain a

trace of one memory can be overwritten when other memories are stored as synapses might be shared among different memories. They explored if the performance on memory retention is improved by allowing synapses to traverse a large number of states before reaching their bounds. They analysed two cases: hard bounds and soft bounds. In the case of hard bounds, potentiation or depression are constant and they do not depend on the synaptic weights. In this case they demonstrated that, when there is a balance between excitation and inhibition, memory lifetimes grow proportionally to the square of the number of synaptic states. In the case of soft bounds, in which the probability of weight change depends on the synaptic strength, memory retention is improved without the need of fine tuning network parameters. This improvement has been shown to be linear with respect to the number of synaptic states.

6.5 Discussion and Conclusions

We demonstrated that it is possible to control recurrent networks in neuromorphic devices. In particular, we applied a mean-field theory guided approach to choose a set of parameters that has been successfully mapped in hardware. The effective transfer function together with the experimental protocol to measure it, have been explained. This measure represents a way to calibrate the massive feedback loops that give rise to discrete meta-stable states in the network dynamics. In addition we demonstrated, starting from a uniform network composed of spiking neurons and Hebbian-like spike-driven synapses, that it is possible to autonomously develop attractor representations of input stimuli. The attractor dynamics ensure that after the removal of the stimulus, there is reverberant elevated activity. The problem of network capacity has been discussed in Section 6.4. Networks of neurons with the implemented learning dynamics, and in the case of hard-bounded synapses have a memory capacity that scales as the logarithm of the number of synapses [Fusi, 2002], while it scales linearly in the case of Hopfield networks [Hopfield, 1982]. This limitation can be partially defeated by relying on stochastic updates, or by simply lowering the coding level [Fusi and Abbott, 2007]. However a quantitative comparison with the theoretical estimates is still missing. This is caused by several factors. Firstly, the fact that the probability of weight update is affected by mismatch in the analog synaptic circuits, this creates a distribution of probabilities across the network. Secondly, the probability changes are not constant during the learning history because mean rate frequencies depend on synaptic states, and these states changes during learning. Thirdly, finite-size effects influence synaptic dynamics via the peculiar distribution of firing rates, resulting in strong biases towards potentiation or depression. The choice of a low coding level is essential when working with small scale recurrent networks. This choice has been made in order to average synaptic mismatches across the dendritic tree of

each neuron in the neuromorphic processor. We focused on learning orthogonal patterns, this means patterns in which the interference induced on ongoing synaptic changes by different stimuli is minimal. However the stop-learning mechanism could be exploited in a more realistic situation in which synaptic changes caused by different patterns can in part overlap. This scenario has been already explored in a perceptron network realized in neuromorphic hardware [Giulioni et al., 2009]. The use of a stop-learning mechanism would provide a valid option for the development of attractor networks in the case of more natural input stimuli. Since attractor models are considered relevant components in a variety of computational scenarios in neuro-biological networks, the presented work constitutes a significant step towards biologically-inspired distributed information processing systems.

CHAPTER 7

Decision making and perceptual bistability in spike-based neuromorphic VLSI systems

F. Corradi, H. You, M. Giulioni, and G. Indiveri

International Symposium on Circuits and Systems, ISCAS, 2015

Much effort is currently being invested in the quest for developing new computing paradigms for information and communication technologies (ICT): fundamental notions are being revised and fundamental characteristics of new materials are being explored to develop new types of computing systems that can go beyond the Complementary Metal–Oxide–Semiconductor era. In this work we address this challenge by taking inspiration from the efficiency and robustness of neuro-biological systems: we study an implementation of a brain-inspired model of basic computational primitives which uses low-power mixed signal subthreshold analog and asynchronous digital circuits to implement a network of spiking neurons and synapses. Despite the variability and heterogeneity observed in the analog circuits, we demonstrate a reliable neuromorphic implementation of neural processes involved in the foundations of bistable perception, decision making, and working memory.

7.1 Recurrent attractor networks as a model for decision making and perceptual bistability

7.1.1 A mean rate model

Describing the neural dynamics of perceptual discrimination with mean rate models can provide insight on the neural processes involved during the decision task. In a work done by [Wong and Wang, 2006], a reduced two variable neural model is presented. This model offers a simple and biologically plausible framework for studying the perceptual decision making task. The work is based on a two populations network model in which the network dynamics is studied in a mean field approximation. In this work, they present a two variable system of equations and the study of the phase-plane elucidate the dynamical mechanisms underlying network behavior. In our analysis, we will compare the phase plane obtained by the theoretical model with the decision space obtained from our neuromorphic spiking analog system. The system of equations provided by the model is:

$$\begin{aligned} ds_a &= -\tfrac{s_a}{\tau_s} + (1 - s_a)\gamma f(I_{syn}, a) \\ ds_b &= -\tfrac{s_b}{\tau_s} + (1 - s_b)\gamma f(I_{syn}, b) \end{aligned} \tag{7.1}$$

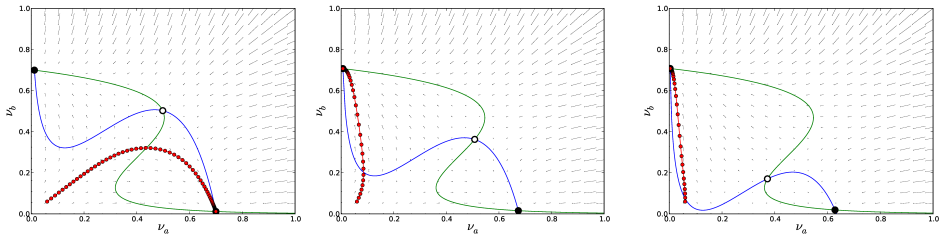
Where s_a, s_b are describing NMDA gating variables for the two populations of neurons (A, B). The function $f(I)$ represents the population firing rate f-I curve, which describes the input output relation for the two populations of neurons. I_{syn} is the input synaptic current and depends from the connectivity in the network c_{ij}, and a constant injection of current I_0 as follow:

$$\begin{aligned} I_{syn,a} &= c_{11}s_1 - c_{12}s_2 + I_0 + I_1 \\ I_{syn,b} &= -c_{21}s_1 + c_{22}s_2 + I_0 + I_2 \end{aligned} \tag{7.2}$$

I_1, I_2 define input currents to the two populations that are caused by the motion stimuli. The function $f(I)$ defines how neurons' firing rate depends on its input current and it is defined as

$$f(I) = \frac{x - Iy}{1 - exp(-d(xI - y))} \tag{7.3}$$

in which x,y and d are constants whose values are $x = 220Hz/nA$, $y = 108Hz$ and $d = 0.15s$. These constants shape neuron's response function. The study of the phase-plane portrait for the mean-field network evidences stable points in the dynamics as shown in Fig. 7.1. In this figure continuous lines represent nullclines for the systems and white dots unstable point of the dynamics. Black dots evidence stable points in the network dynamics, i.e. attractors states. In Fig. 7.1a the phase plane shows when the two populations of neurons (A, B) are receiving same inputs. In this case it is found an unstable point along the bisect of the plane, this gives the equal probability to the two



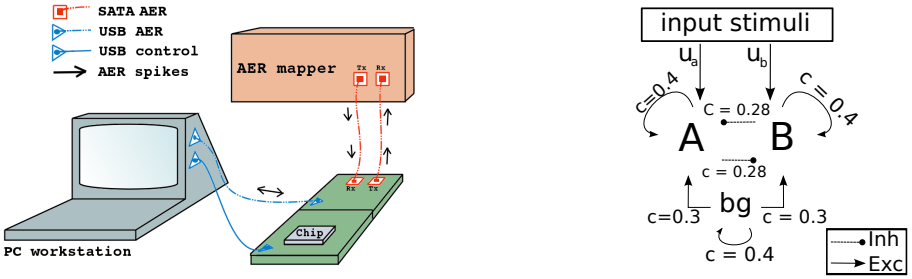
(a) (b) (c)

Figure 7.1: **Decision space for mean rate model**. Continuous lines depict nullclines for the system. White dot identify unstable point of the dynamics. Black dots evidence stable point of the network dynamics. Red dots evidence a typical trajectory of a single trial. a Phase space when input coherence is 0%. The unstable point is found at the bisect. b Input coherence 10%. The unstable point is moved from the center and the probability of choosing option B increases. c Input coherence 100%.

choices. In red dots is plotted a single trial or trajectory in which network activity is finally found in attractor A. The comparison of the behavior of the neuromorphic system, visibile in Fig. 7.4, with the phase space obtained by the system of equations 7.1 shows that the two systems shares the same behavior. The parameters used to produce Fig. 7.1 are: $\tau_s = 0.06$, $\gamma = 0.65$, $x = 270Hz/nA$, $y = 108Hz$ and $d = 0.156s$.

7.1.2 Network architecture

The attractor network is composed of adaptive-exponential integrate-and-fire neurons. The network is populated with excitatory and inhibitory pools of neurons coupled with local excitation and global inhibition. The balance among excitation and inhibition is used to produce a competitive winner-take-all dynamics, and it is a common architecture in decision-making and persistent activity circuits. The architecture of the network we have implemented is illustrated in Fig. 7.2b. Network architecture is made of four populations of neurons, A, B, and Back. The excitatory populations (A, B, bg) are recurrently connected with sparse connectivity and inhibit each other via direct inhibitory connections. The number of neurons in population A and B is $N_a = N_b = 22$ while the background population counts $N_{bg} = 12$ neurons. Connectivity levels in the network are random and sparse. Connectivity c in Fig. 7.2b refers to the mean number of connections that a presynaptic neuron has with a post-synaptic neuron randomly chosen in the target population. The background population is used as a source of excitation for populations A and B. In the network, the spontaneous activity of neurons is enough to trigger meta-stable state transitions and phase space explorations.



(a) (b)

Figure 7.2: a The Neuromorphic setup. b The network consists of three populations of neurons (A, B, bg) recurrently connected. The connectivity in the network is random and sparse. Connectivity levels are as indicated in the diagram.

7.1.3 Neuromorphic setup

We use a neuromorphic system to study recurrent network dynamics. The setup is composed of a standard workstation, a silicon neuromorphic chip that contains 58 adaptive exponential integrate-and-fire neurons, and a mapper board (Fig. 7.2a shows a schematic representation of the full setup). The chip is a mixed signal analog/digital custom design. It has been fabricated with a 0.18 μm complementary metal oxide semiconductor (CMOS) 1-poly 4-metal technology. The neuron model implemented in the chip is a two-variable exponential adaptive model described in [Brette and Gerstner, 2005]. The circuit implementation can be found in [Indiveri et al., 2011] under the DPI adaptive exponential neuron model. The mapper board is connected with the chip via a standard SATA cable. The role of the mapper is to store a synaptic matrix, and its function is completely deterministic. The mapper only serves to route spikes back to the chip and/or to the computer. The neuromorphic chip is connected to a daughter board (AMDA), which in turn is connected via a USB cable to the computer. The AMDA board only serves to configure the analog parameters of the synapses and neurons in the network.

7.1.4 The chip architecture

The neuromorphic chip comprises five main blocks: the asynchronous input/output interfaces, the SRAM block, the learning bistable plastic synapse block, a programmable de-multiplexer, and the neuron array block. The asynchronous input/output interfaces are custom made encoder/decoder circuits and are used to manage the communication among external digital asynchronous signals and the signals internal to the chip [Mostafa et al., 2013]. In fact, spikes are communicated digitally and asynchronously to/from the

chip. The SRAM block is used to store synaptic weights with four bit resolution in a 10T low-leakage sram cell [Lo and Huang, 2011]. The SRAM block generates a dual-rail input/output data scheme by means of a filter circuit previously introduced in [Moradi and Indiveri, 2011]. The synaptic weights are converted into currents by DAC synapse circuits. These currents are then fed in one of the two DPI synapses circuits, depending on the digital input address. On chip there are two distinct DPI circuits: one is used to produce excitatory post synaptic potential while the second is used to produced inhibitory post synaptic potentials. The current generated by the DPI circuit is then sent to the adaptive exponential integrate and fire neuron. If instead of the SRAM block, the input spike is addressed to the plastic bistable learning synapse array, the input spike is first directed into the synapse shaper that extends the pulse of few microseconds. This pulse is used to generate a post synaptic excitatory potential if the synapse is found in the up state. The synapse can go under potentiation or depression depending on the activity of pre and post synaptic neurons. The learning rule and the details of this implementation can be found in [Fusi et al., 2000].

7.2 Results: Discrimination of distinct stimuli and basins of attraction

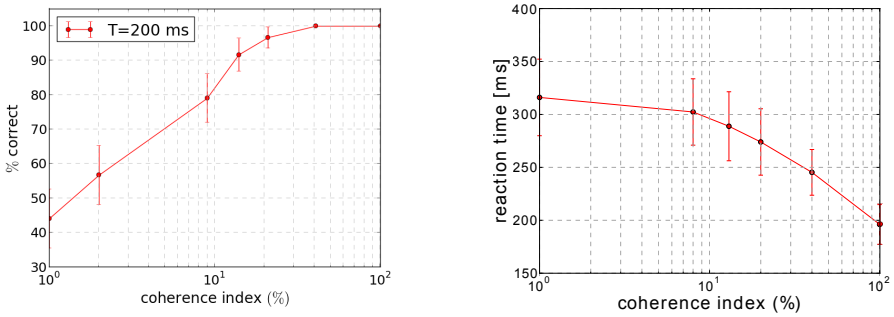
Long-lasting spiking activity in the cortex, that lasts much longer than the typical time scales of synapses and membrane potentials, is the neural correlate of working memory and perceptual decision making [Wang, 2002]. In theoretical neuroscience models this activity is assumed to encode sensory inputs which relate to different alternative possible decisions or percepts. This activity is accumulated in different pools of neurons that, due to their recurrent excitatory connections, are capable of sustaining persistent activity, even after the input stimulus that triggered it is removed. Previous theoretical work has demonstrated that neural circuits of the type shown in Fig. 7.2b can account for salient characteristics of the neural correlates of perceptual decision making such as psychometric functions and reaction times [Wang, 2002; Wong and Wang, 2006].

We emulated a two choice discrimination task using the two pools of silicon neurons described in Section 3.2.1. In particular, we reproduced the results of a classical neuroscience experiment denoted as the two alternative random moving dots task [Glimcher, 2003]. In this experiment, monkeys or human subjects have to report the net direction of perceived motion (left or right) of a cloud of moving dots on a screen. The experiment consists of many trials in which it is changed the amount of coherently moving dots in one direction, versus the amount of dots moving in random directions. While performing the task, subjects accumulate evidences for a decision and report the perceived direction of motion as quickly as possible. The decision process is triggered when the accumulation

of evidence reaches a threshold. The speed of execution of the task depends on the motion coherence (percentage of coherently moving dots in a given direction). In our experiment we bypassed the visual processing stages and stimulated the populations of neurons representing the perception of moving dots directly with computer generated spike trains. In particular, during the stimulation phase, we stimulated both populations A and B with inhomogeneous Poisson spike trains that represent the activity of the middle temporal (MT) visual area during a random moving dots task. In addition, we stimulated all neurons chip with a Poisson spike train of $10\,Hz$ to represent background activity. The mean rates of the input stimuli are expressed as:

$$\begin{aligned} \nu_a^i &= \nu_0 - \alpha \cdot \nu_{coh}^i \\ \nu_b^i &= \nu_0 + \alpha \cdot \nu_{coh}^i \end{aligned} \tag{7.4}$$

where ν_0 is the base stimulation frequency, α represents a ramping coefficient, ν_{coh}^i represents the percentage of motion coherence, and i indicates the experiment trial. The coherence factor ν_{coh}^i is limited in the range: $0 < \nu_{coh}^i < 100$. Therefore, if ν_{coh} is large, the two populations will receive largely different inputs: ν_a will be large and ν_b will be small. The competition between the two neural populations will eventually collapse in one of the two attractor states. The persistent activity that remains after the input stimulus removal, will be sustained by the recurrent network dynamics. This mnemonic delay period is the neural correlate of working memory. The choice of the network is probed by means of a threshold on the mean firing rates ($\nu_{thr} = 50\,Hz$): when one of the two population exceed this threshold than we assume that the network dynamics has committed to a decision. The time required for the two pools of neurons to finish the competition represents the reaction time of the trial. For trials in which the coherence is equal to zero, the two input stimuli have the same mean frequencies, which indicate a non informative input. While for coherence levels close to 100%, the mean input frequencies are about $\nu_a = 100\,Hz$ and $\nu_b = 0\,Hz$ indicating a complete coherent stimulus. The decision space can be well represented by a 2D plot in the frequency space ν_a, ν_b. Such a figure describes the time spent by the two populations of neurons at different firing rate frequencies. In Fig. 7.4 we show three different cases in which the input stimulus was at different coherence levels. On the x-axis we plot the mean firing rate frequency for population A, and on the y-axis we plot the firing rate for population B. Every plot in Fig. 7.4a, 7.4c, 7.4e is the average over 300 trials. At the beginning of the trial, the network dynamics is slowly moved from the spontaneous activity state ($\nu_a \sim \nu_b \sim 5\,Hz$) to a point in which both populations are firing at higher rate. In this phase, the rate of the two populations strictly depends on the input stimuli and the network is integrating evidences for the subsequent decision. After 200 ms from the presentation of the stimulus,



(a) (b)

Figure 7.3: **Neurometric functions.** Every dot represents the mean of 300 trials. Error bars are the standard deviations. As the coherence level increases the performances increase. Moreover, the reaction time decrease with an increase coherence of the input stimuli which represents an easier discrimination trial.

the input is removed from the network and the network dynamics collapses in one of the two attractor states A or B. The collapse in one of the two attractor state is the neural correlates of a decision. When the mean firing rate activity is higher for population A (or B) then the network choice is A (or B). In Fig. 7.4a is shown the decision space for trials in which the coherence level was 0. This means that the network had equal information about the inputs and the choice was made at chance level ($\sim 50\%$). In Fig. 7.4b ten trials are shown in a mean rates plot. As you can see, after the stimulus removal at 0.2s one of the two population smoothly switches off while the other increases its firing rate. Note that it is not possible to end with both populations firing at elevated reverberant state as the two pools of neurons inhibit each other. Figures 7.4a, 7.4b evidence that the choice A and B where almost equiprobable and the network was performing at chance level. When the coherence of the input stimulus increases the network performs better than chance level. For high coherence of the input, the network is always making the correct decision (see Fig. 7.3). Figure 7.3 shows the psychometric functions for six different levels of input coherence. Every dot in the psychometric function is the average over 300 trials. The reaction time indicates the difficulty of the trial. In fact the reaction time decreases for higher coherence of the input stimuli while for low coherence stimuli it saturates around 320 ms, visible in Fig. 7.3b. The fine tuning of the reaction time can be achieved by driving, with the external input, the system in the proximity of the bifurcation value [Wong and Wang, 2006]. This tuning has been exploited in order to achieve biological realistic reaction time as indicated in Fig. 7.3b.

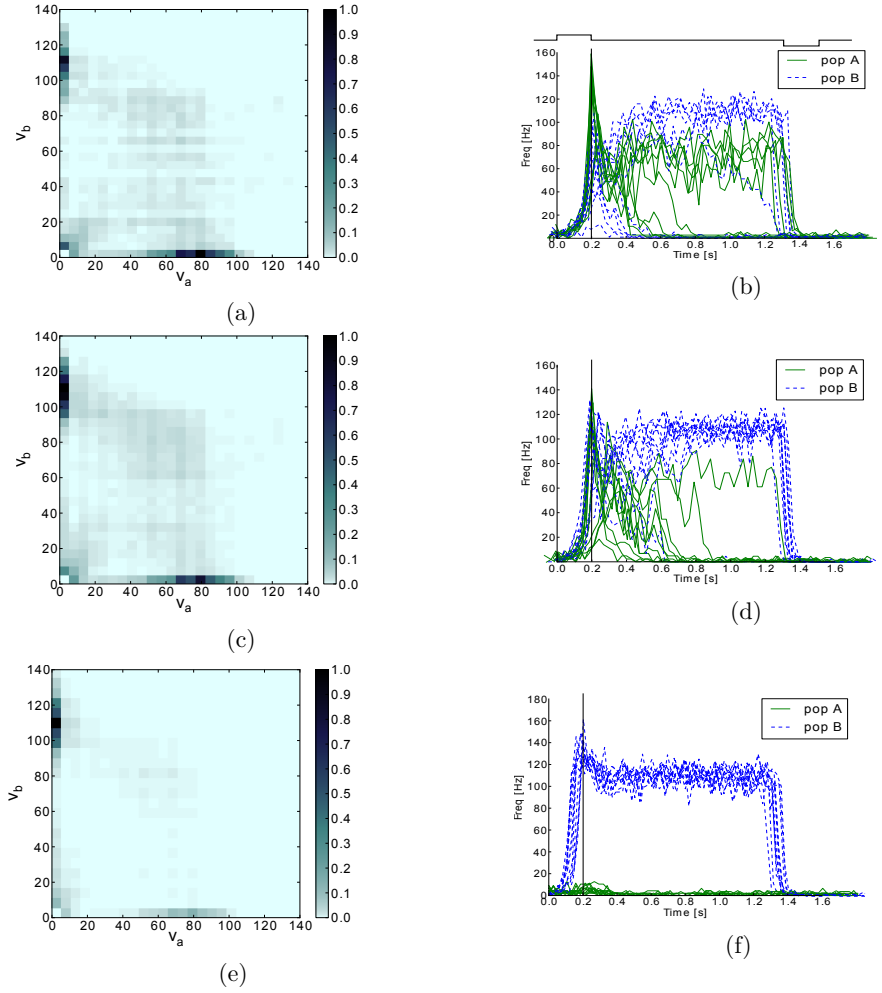
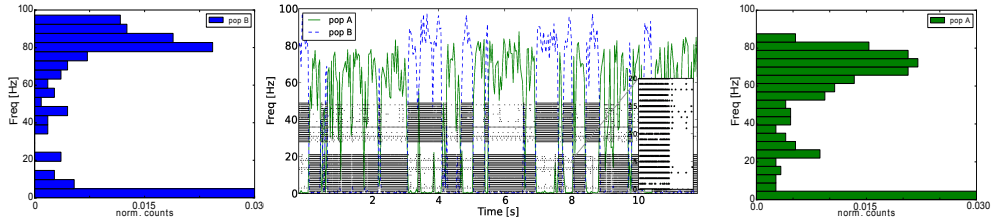


Figure 7.4: **Decision space and mean rate plots.** a, b: $\nu^1_{coh} = 1$ %, c, d: $\nu^2_{coh} = 20$ %, and e, f: $\nu^3_{coh} = 100$ %. Plots a, c, e represent the average of 300 trials and they show the averaged network activity, i.e. the mean rate frequency of populations A and B. Figures. b, d, f show the mean firing rates for ten trials for different coherence levels. The square wave at the top of Fig. d indicates that the stimulation phase lasts $200ms$ and that at $1.4s$ an inhibitory stimulus is used to suppress reverberant network activity. After the stimulus removal $t = 0.2$, network activity collapses in one of the two attractor states. The density for the A choice decrease as the coherence level for the opposite stimulus increase. This is evident from the smaller blue central blob that gradually decrease in size, from 7.4a, 7.4c, to 7.4e.



(a) (b) (c)

Figure 7.5: **Bistable dynamics.** Populations mean firing rate over time. The input to the network is a constant injection of current in the soma of all neurons.

7.2.1 Dynamics of perceptual bistability

When the recurrent connections among neurons that are part of the same population (A, B in Fig. 7.2b) are strong enough, the network operates in a winner-take-all regime. In this regime, only one excitatory population can be in the attractor state at any point in time. Point-attractor neural networks have two types of stable fixed points of the network dynamics. They exhibit a spontaneous state with a low firing rate (down state), and one or more persistent states with high firing rates in which the activity of the network tends to be stable (attractor or up state). The system will react to external stimulation, i.e. to a destabilizing stimulus, with different patterns of activations but it will always relax toward one of the stable attractor state [Haake et al., 1981]. In a winner-take-all regime, since the two populations inhibit each other, only one population of neurons can be found in the up state of the network dynamics. The observed behaviour of our neuromorphic attractor network, when all neurons are stimulated by a constant injection of current, is an alternation of perceptual dominance among the two different activity states with very long time constants, orders of seconds. Figure 7.5 shows five seconds of recordings of such behaviour. The central panel of Fig. 7.5b, shows the mean firing rate activity over time of neurons grouped in populations. Continuous green line depicts neurons in population A, and the dashed blue line represents neurons in population B. An irregular alternation of high activity is evident, and only one of the two populations of neurons can be found in the up state ($\sim 80\,\text{Hz}$) of activity: this is a confirmation that the network is operating in a winner-take-all regime. In conclusion, in this experiment we demonstrated an implementation in neuromorphic hardware of basic computational primitives used to produce neural plausible dynamics. In particular, our results well correlate not only with the behavioral responses (the reaction time and the accuracy) of subjects, but also to the neural responses in their cortical areas during a two-choice perceptual discrimination task. Choice formation and successive behavior coincide with the transition from a spontaneous activity state to the elevated persistent

state of activity in a clustered pool of neurons. An important aspect of this work is that reliable computation emerges from simple mismatched analog neurons; mismatch effects are evident in the mean rate plots, of Fig. 7.5, 7.2, where the frequency of the up states differ of about 10%.

7.3 Long term behaviour and unpredictability of state transitions

We are interested in knowing whether the long term behavior, the succession of the states, might be predictable. The predictability of this alternation might underlie the mechanisms of the perceptual switches. In fact, the nature of this perceptual alternations is still an open debate. In many models, alternations are caused from some form of synaptic adaptation that acts on the dominant population, this leads to a decrease in mean firing rate of the active population, and a successive switch in the perceptual state [Lago-Fernández and Deco, 2002]. These models, in the absence of noise or finite network size effects, produces perfectly periodical switches. Alternative hypothesis state that alternations are mainly caused by noise (external, internal, or both). In the brain, noise is unavoidably present at all levels. From ion channels, where the probabilistic gating of voltage-dependent ion channels causes a natural source of noise. Neurotransmitters are in fact released in quanta, the content of synaptic vesicles, even without neural stimulation [Katz and Miledi, 1972; Lisman et al., 2007]. To single cells, where it is always the case that there is a trial to trial variability in neuronal recordings even when stimuli are identical. For this reason, we estimate the largest Lyapunov exponent of the time series of the perceptual switches. This exponent provides an estimate of chaos in a purely deterministic system by quantifying the exponential divergence of initially close state space trajectories. Chaos is aperiodic long term behavior in a deterministic system that exhibits sensitive dependence on initial conditions. Aperiodic long term behaviour means that there are trajectories which do not settle down to fixed points. Deterministic means that the system has no random or noisy inputs. Irregular behaviour arise purely from the system dynamics in a finite size network. In order to obtain such quantification we first encode the activity of the network in a time series. We then applied standard non-linear time series analysis methods. We use a delay coordinate embedding method for reconstructing the attractor of the system from the observed time series [Takens, 1981]. For this purpose, we first estimate the proper embedding delay with the mutual information method [Fraser and Swinney, 1986]. We determine the embedding dimensions by exploiting the false nearest neighbour method [Kennel et al., 1992]. In what follow we will provide all details of our analysis.

7.3.1 Message encoding

As a fist step, we proceeded by transforming the network state output into a word, that is a sequence of finitely many symbols. We analysed the signal with the temporal time bin coding [MacKay and McCulloch, 1952] in which the output firing rate of the two populations (A, B) of neurons is considered. Let assume we start recording the spiking output from all neurons in the network at time t_0 and at time $t_0 + T$ we stop our recordings. We then split the time interval $[t_0, t_0 + T]$ into n bins $[t_{i-1}, t_i]$ $(i \leq i \leq n$, with $t_n = t_0 + T$) of same length $\Delta\tau = \frac{T}{n}$. Each bin is now coded with $0, 1, -1$ according to whether the network state have been found in the lower activity state (average mean firing rate activity of both populations $< 10Hz$), or in the self-reverberant state for population A (mean firing activity of population A $> 15Hz$) or else population B would have been found in its meta-stable self-reverberant state of activity (mean firing rate pop. B $> 15Hz$). The result is a long sequence $s_i, .., s_n$ which only contains three distinct symbols: $0, 1, -1$. With that sequence of symbols we produce a *message* by creating an array of values $a_i, .., a_m$ in which each element is calculated as follow:

$$a_i = \sum_{s_i}^{7} 2^i \tag{7.5}$$

In Fig. 7.7 we show the result of this *encoding*. Since we are summing over 8 bits ($i = 0, ..., 7$ in eq. 7.5) the value of the message is bounded between -128 and 128. The message, for obvious reasons, can be view as emitted by a neuronal information source S [Amigó et al., 2004]. The result of this encoding is a single time series of a single variable (a). In order to quantify the information contained in the time series, we will reconstruct this sequence as best as possible.

7.3.2 Reconstruction of the attractor

According to [Takens, 1981], it is possible to reconstruct the attractor in a m-dimensional space of the original system. This can be achieved by organizing the observed system states in a sequence

$$p_i = \{x_i, x_{i+\tau}, x_{i+2\tau}, ..x_{i+(m-1)\tau}\} \tag{7.6}$$

where τ and m are the embedding delay and the embedding dimensions of the system. The reconstructed attractor has the same mathematical properties as the original one, such as the dimensions, Lyapunov exponents, etc. Even if it might seems abstruse to reconstruct the attractor from only a single time series, in [Kennel et al., 1992] the authors give a very intuitive explanation that lies in the fact that all variables in a non-linear system influence one with each other. Therefore the result of a single measurement is an

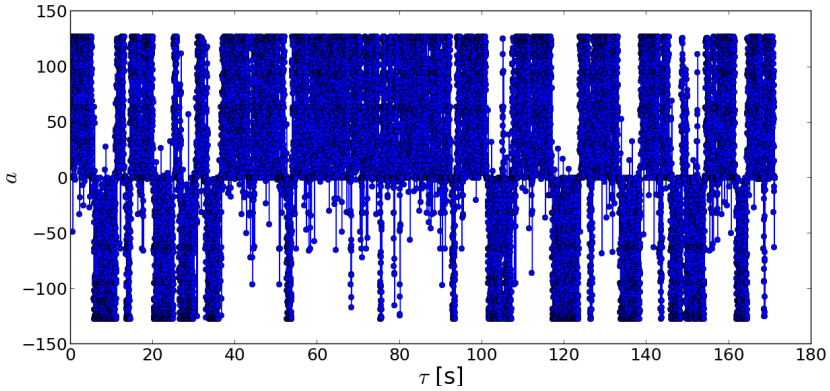


Figure 7.6: **Time series emitted by the neuronal source**. i refers to the step of encoding while a is the encoded value as defined in eq. 7.5

entangled combinations of all the system variables. Following this reasoning, $x_{i+\tau}$ might be interpreted as a substitute second system variable, which carries information about the influence of all other variables during time τ. Accordingly, we could introduce the 3rd ($x_{i+2\tau}$), 4th ($x_{i+3\tau}$), ..., mth ($x_{i+(m-1)\tau}$) substitute variables, and therefore obtain the full m-dimensional phase space in which the substitutions contain all influences of the original system variables. This is consistent only if m in eq. 7.6 is large enough. The use of equation 7.6 is possible only if an appropriate value for m, and τ is determined. In order to determine the value of m, we used a method introduced by [Fraser and Swinney, 1986]. In this method, the mutual information is used to estimate time delay τ in phase-portrait reconstruction from time-series data. The mutual information between x_i and $x_{i+\tau}$ quantifies the amount of information we have about the state $x_{i+\tau}$ presuming we had access to x_i. In order to calculate the mutual information for a time series of the form $\{x_0, x_1, ..., x_i, ..., x_n\}$, we have to first find the minimum x_{min}, the maximum x_{max}, and the absolute value of their difference $|x_{min} - x_{max}|$. At this point, we partition the absolute value of the difference into j equal intervals and we calculate the expression:

$$I(\tau) = -\sum_{h=1}^{j}\sum_{k=1}^{j} P_{h,k}(\tau) ln \frac{P_{h,k}(\tau)}{P_h P_k} \tag{7.7}$$

in which P_h, P_k are the probabilities that the variable assumes a value inside the hth and the kth bins, respectively. While $P_{h,k}(\tau)$ is the joint probability that x_i is in bin h and $x_{i+\tau}$ is in bin k. In case of chaotic behaviour we have that $I(\tau) \to 0$ as $\tau \to \infty$, there is no correlation between x_i and $x_{i+\tau}$. In fact in eq. 7.7, $P_{h,k}(\tau)$ factorize to $P_h P_k$

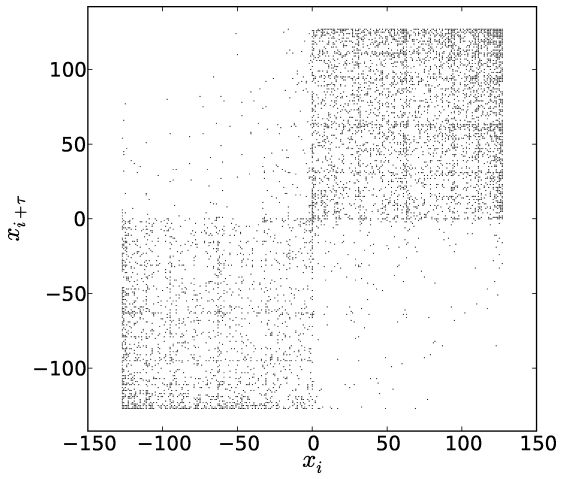
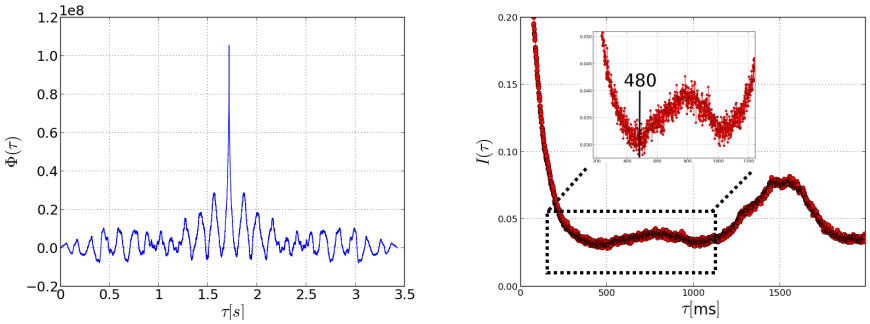


Figure 7.7: **Attractors reconstruction**. The delay is equal to $\tau = 480$ ms. This value corresponds to the first minima of the mutual information (see also Fig. 7.8b).

resulting in a zero. We are interested in the minima of $I(\tau)$. At the first minimum, $x_{i+\tau}$ adds the larger amount of information we had by knowing x_i without loosing all the correlation between them [Fraser and Swinney, 1986]. Therefore the first minimum of $I(\tau)$ represents the best choice for the value of the embedding delay τ. Fig. 7.8b shows the mutual information $I(\tau)$ for the encoded time series. The inset evidence the first minima, it is found at $\tau = 480ms$. Other methods, uses as approximation of τ, the value of the autocorrelation function where it drops to $1 - 1/e$ of its initial value [Rosenstein et al., 1993]. Fig. 7.8a shows the autocorrelation function of the time series. As it can be seen in Fig. 7.8a, the autocorrelation drop to zero as time increases. To determine the embedded dimensions m, we use the method called false nearest neighbors [Kennel et al., 1992]. The method exploits the fact that points that are close in the reconstructed embedding space have to stay sufficiently close also during the next iteration. After some iterations, the distance between two points p(i) and p(j) of the reconstructed attractor, which initially where only at distance ϵ, cannot grow further. Any point ith that has a close neighbor that does not fulfill this criterion is said to have a false nearest neighbor. By choosing m large we are minimizing the fraction of points that have a false nearest neighbor. The algorithm can be summarized in the following steps:

- Given a point $p(i)$ in the time series, look for its nearest neighbour $p(j)$ such that $\|p(i) - p(j) < \epsilon\|$, ($\epsilon$ is a constant not larger than the standard deviation)
- The normalized distance R_i between the $(m+1)th$ embedding coordinate of points



(a) (b)

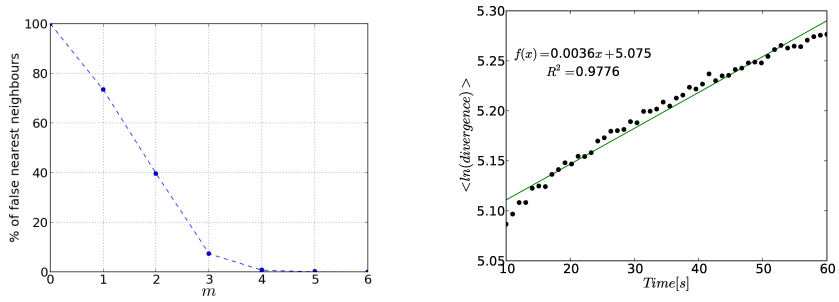
Figure 7.8: **Embedding delays estimation.** a) Autocorrelation of the time series. b) Mutual information $I(\tau)$. The first minima in the mutual information is found for 480 embedded delays. In the inset, the black vertical line highlight the minima.

$p(i)$, and $p(j)$ is calculated as

$$R_i = \frac{\|x_{i+m\tau} - x_{j+m\tau}\|}{\|p(i) - p(j)\|} \tag{7.8}$$

7.3.3 Estimation of the largest Lyapunov exponents

For time series produced by a dynamical system, sensitivity to the intial conditions is characterized by the Lyapunov exponents. In a chaotic dynamical system, trajectories with nearby initial conditions will diverge, on average, at exponential rate characterized by the Lyapunov exponents [Eckmann and Ruelle, 1985]. For the above system, it is reasonable to consider a 4D sphere of initial conditions. As time progresses, the sphere would evolve into an ellipsoid whose principal axes expand or contract at a rates which are proportional to the Lyapunov exponents. For diagnostic chaotic behaviour, it is sufficient the presence of a single, i.e. along one dimension, positive Lyapunov exponent. The information created by the system is proportional to the change in volume defined by the expanding principal axes. To obtain such estimation, we used an algorithm proposed by [Rosenstein et al., 1993]. The algorithm relies on a single time series of measurements, and also works for systems where the system formulas are not available. The method starts by reconstructing an approximation of the system dynamics by embedding the time series in a phase space where each point is a vector of the previous m points in time, each separated by a lag τ. If we examine each point of the embedded time series, we can find its Euclidean nearest neighbor whose temporal distance is greater that the mean period of the system. This corresponds to the next cycle in the system's attractor. The algorithm considers each pair of neighbors as nearby initial conditions for different



(a) (b)

Figure 7.9: 7.9a **Minimum embedding dimension estimation with false nearest neighbors**. The percentage of false nearest neighbors clearly drops to zero in a 4D system. 7.9b **Largest Lyapunov exponent estimated with Rosenstein's algorithm.** Continuous line shows the linear fit of the data (dots). The positive slope indicates a positive largest Lyapunov exponent 0.0036.

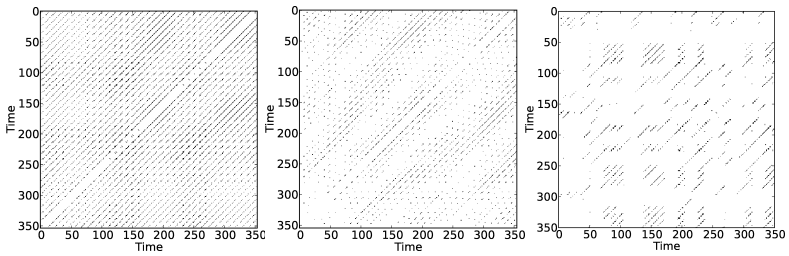
trajectories. The largest Lyapunov exponent is then estimated as the mean rate of separation of the nearest neighbors. The process is then repeated for each step in time for all pairs of neighbors. The results is an average estimates of separation. The fact that these estimates are averaged over multiple trajectories over the entire time series, allow for an accurate and fast results even in presence of noise. Fig. 7.9b we show the averaged logarithm of the separation.

7.3.4 Recurrence quantification analysis

Recurrence plot is a technique first introduced in [Eckmann et al., 1987]. This technique is used to visualize the recurrence behavior of a dynamical system embedded in a phase space. A recurrence plot is constructed using the following equation,

$$R_{i,j} = \theta(\epsilon_i - \|x_i - x_j\|) \tag{7.9}$$

where x_i represents the point in phase space in which the system is at time i, ϵ_i is a threshold, and θ is the heavy-side function. The matrix $R_{i,j}$ consists of zeros and ones and is usually represented as a black and white plot. Ones, or black dots mean that the system returns to the proximity (ϵ_i neighborhood) of the corresponding point in the phase space. The plot quantifies the number and duration of the recurrences of a dynamical system [Marwan et al., 2007]. In Fig. 7.10 we show the recurrence plots of the time series (a), as defined in eq: 7.5. The three plot are time series of data for different coherence levels. When input coherence is 100% (see Fig. 7.10a), the system evidence a deterministic dynamics, as indicated by the continuous diagonal lines. When input



(a) (b) (c)

Figure 7.10: **Recurrence plots.** a Coherence of the stimulus was 100%. Continuous diagonal lines indicates a deterministic oscillatory behavior, i.e. the perceptual decision is always correct. b Coherence was reduced to 10%. Diagonal lines are partially disrupted, indicating the presence of stochasticity in the decision. c Coherence level 1%. Small scale structures emerge and diagonal lines are highly disrupted, which is a strong indicator of unpredictable behaviour of the system. All plots are averages over 40 trials.

coherence is decreased to 10% (see Fig. 7.10b), the diagonal lines are partially disrupted, indicating the presence of noise or stochasticity in the system dynamics. Finally if input coherence is reduced even further (see 1% in Fig. 7.10c) small scale structures emerges and diagonal lines are highly disrupted, this is a strong indicator that the system is going trough some complex stochastic or noisy dynamics [Eckmann et al., 1987].

7.4 Discussion and Conclusions

The capability of embodying information in the dynamics of a recurrent network in the absence of external stimulation with the consequential transitions between meta-stable states represents a fundamental processing capability of neural systems. In this chapter, we have presented a neuromorphic implementation of neural processes involved in the foundations of mental representations. In particular we analyze, in a network of spiking neurons, processes that are at the basis of bistable perception, decision making, and working memory. The balance among excitation and inhibition is used to produce a competitive winner-take-all dynamics, and it is a common architecture in decision-making and persistent activity circuits. We demonstrate that our neuromorphic analog system generates dynamics capable of reproducing both neurophysiological data and psychophysical performances in a perceptual discrimination task between two alternatives. Moreover, we analyzed the long term behavior of the network dynamics when the network is performing a difficult choice within two alternatives. We show that the network dynamics exhibits stochastic switching among different alternatives. We calculate the largest Lyapunov exponent of the time series produced by the stochastic switching of the meta-stable states. The result is a positive largest Lyapunov exponent, which is

a strong indicator of the unpredictability of state transitions. We also confirm the unpredictability of state transitions by studying the attractor phase space by means of recurrence quantification analysis techniques.

An important aspect of this work is that reliable computation emerges from simple mismatched analog neurons; mismatch effects are evident in the mean rate plots, Fig. 7.5, 7.2, where the frequency of the up states differ of about 10%.

CHAPTER 8

Computing with the Neural Engineering Framework

F. Corradi, C. Eliasmith, and G. Indiveri

International Symposium on Circuits and Systems, ISCAS, 2014

8.1 Theory

Several promising neuromorphic devices have recently proposed, using both digital and analog design techniques [Arthur et al., 2012; Brink et al., 2013; Cassidy et al., 2013; Choudhary et al., 2012; Moradi and Indiveri, 2011; Qiao et al., 2015; Yu et al., 2012]. However, in order to implement full-fledged computing systems, starting from these types of devices, it is necessary to adopt a formalism that can best exploit the properties of such computing elements. The NEF [Eliasmith and Anderson, 2003] represents a synthesis of multiple approaches in computational neuroscience, computer science, communications and control theory, that can provide a formalism to build large-scale networks models. This computational framework has been previously introduced in [Eliasmith and Anderson, 2003] and it has been used to simulate in Software (SW) large spiking neural networks capable of reproducing many aspects of neural systems ranging from the neurophysiological level all the way up to the behavioral one [Eliasmith et al., 2012]. The neural engineering framework (NEF) [Eliasmith and Anderson, 2003] is based on control theory and integrates three basic principles:

8.1.1 Representation

A stimulus $x(t)$ is encoded as spiking activity $a_i(x(t))$ by a pool of neurons with different transfer characteristics:

$$a_i(x(t)) = G_i(\alpha_i e_i x(t) + J_i^{bias} \tag{8.1}$$

where G_i is a spiking neural non-linearity, α is a gain, e is an encoder, and J_i^{bias} is a background current. This activity $a_i(x(t))$ is linearly decoded to retrieve the stimulus $x(t)$,

$$\hat{x}(t) = \sum_i a_i(x(t)) d_i^x \tag{8.2}$$

in which d_i^x represents the decoding weights, and

$$a_i(t) = \sum_n h_{PSC}(t) * \delta(t - t_{in}) \tag{8.3}$$

is the linearly filtered spiking activity. The $h_{PSC}(t)$ is a filter capturing the effects of postsynaptic currents.

8.1.2 Transformation

a stimulus $x(t)$ is transformed into $y(t)$ by a mapping of $a_i(x(t))$ into $b_j(y(t))$. The transformation is a weighted connection between neural pools of neurons that compute a function on the represented value. For example, in the linear case $y(t) = Ax(t)$ is represented by the activity

$$b_j(A\hat{x}(t)) \tag{8.4}$$

In this representation neuron i feeds its output to the input of neuron j by using a weight matrix

$$\omega_{ij} = \alpha_j e_j A d_i. \tag{8.5}$$

8.1.3 Dynamics

Recurrent connections can be computed using the same approach to implement nonlinear and linear dynamical models including attractor networks, Kalman filters, controllable harmonic oscillators, etc. In such a mapping, the standard dynamics matrix A becomes $A' = \tau_{PSC} A + I$, and the input matrix B becomes $B' = \tau_{PSC} B$ where τ_{PSC} is the time constant of the postsynaptic current filter. We applied the NEF to the multi-neuron chip, exploiting its analog silicon neuron properties, and its programmable synaptic weight features.

8.2 Experiment: Mapping arbitrary mathematical functions and dynamical system to neuromoprhic VLSI systems

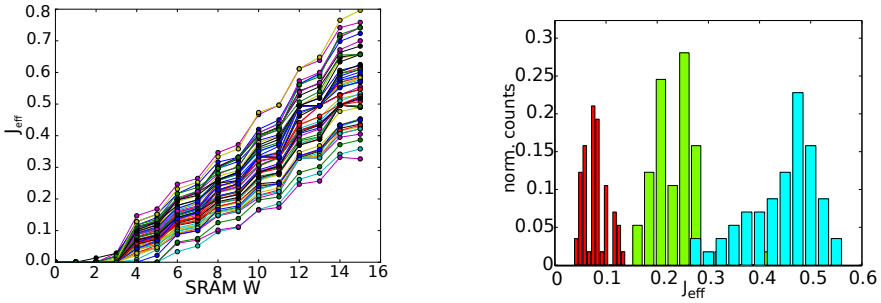
In the next section we validate the NEF by applying its principles to a population of compact low-power silicon neurons, designed using neuromorphic analog circuits and fabricated using a standard 0.18 μm CMOS process (see Chapter 4). We provide experimental results showing the outcome of the calibration procedures required to implement NEF, and of a successful real-time computation of mathematical functions. In addition, using this framework, we construct a dynamical system with two distributed memory states that represent the neural correlate of working memory, and demonstrate its correct real-time performance in Hardware (HW). This is achieved by means of a network of spiking neurons with multiple weighted connections. The synaptic weights are stored in a 4-bit on-chip programmable SRAM block. We propose a parallel event-based method for calibrating appropriately the synaptic weights and demonstrate the method by encoding and decoding arbitrary mathematical functions, and by implementing dynamical systems via recurrent connections.

The neuromorphic device

The VLSI device used in this work is a prototype chip that comprises a network of 58 adaptive exponential I&F neurons, implemented using analog subthreshold circuits. Each neuron has 32 programmable synaptic inputs, with synapse circuits that express biologically plausible neural dynamics. In addition, each neuron has 8 bi-stable synapses, with on-chip plasticity mechanisms. The VLSI chip can receive and transmit pulses representing spikes via asynchronous digital circuits, and following an AER protocol. The chip is connected to a workstation via a USB interface; signals transmitted to the USB bus from the chip encode the address of the source neuron, while signals received by the chip encode the address of the destination synapse. Once off-chip, the spikes produced by the silicon neurons are routed by a "mapper" board, built using a commercial FPGA (Xilinx Spartan-6). The mapper is hosted in a standard workstation and uses the workstation memory to implement a programmable connectivity look-up table with source-destination entries. This setup allows us to construct arbitrary network topologies of spiking neural networks.

8.2.1 Synaptic weights calibration

The asynchronous SRAM synapses have been used to compensate for device mismatch caused by the process variations. One of the main problems, when using analog neuromorphic systems, is the inability to obtain precise synaptic weights. We overcome this



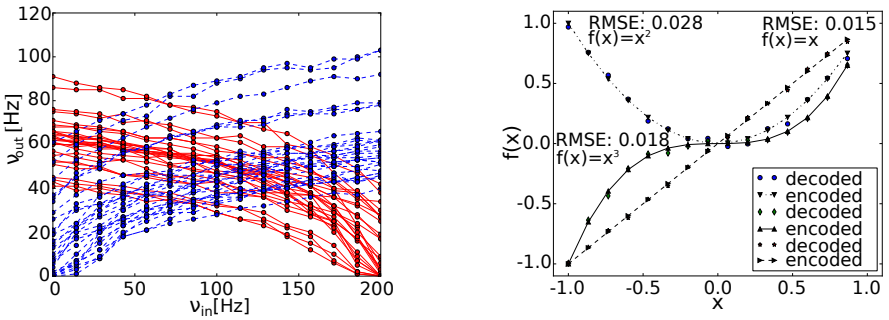
(a) (b)

Figure 8.1: SRAM calibration. 8.1a. Synaptic efficacy (J_{eff}) for different sram weight values. Every line represents a neuron in the array. The synaptic efficacy J is expressed in terms of number of excitation spikes over spike emitted by the neuron under exam. 8.1b. Calibrated SRAM weight to match three different synaptic efficacies.

problem by applying a fast calibration procedure that considers the response function of single neuron by adjusting the digital weight stored in SRAM cells to match for a chosen synaptic efficacy value. The calibration procedure consists of measuring effective synaptic efficacies for all neurons in the array and for all possible digital weights. This procedure can be executed in parallel for all neurons in the chip and it only requires to send (receive) spikes to (from) the chip using the Address-Event Representation. Figure 8.2.1 shows the result of this measure when all neurons are stimulated at 100 Hz for 200 ms. The stimulation is done via synthetic regular spike trains produced by the computer. The complete calibration only takes $200\,\text{ms} * 15\,(\text{bits}) = 3\,\text{s}$. Once we measured all calibration curves, we use the least squares estimates method to estimate a digital value for a given synaptic efficacy. With this method, we are capable of obtaining synaptic efficacies as shown in Figure 8.1b.

8.2.2 Representations of functions with populations of neurons

In general, encoding is obtained by the spiking activity $\delta(t - t_{in})$ of a single neuron i via the nonlinear neuron response function G_i. In practice, encoding thus exploits different tuning curves for neurons that project the stimulus $x(t)$ to a specific neuron space. In fact, tuning curves relates the spiking response of a neuron to a particular stimulus. We show in Figure 8.2a tuning curves for all neurons in the neuromorphic chip. We implemented different bias values J_i^{bias} by stimulating every neuron i with a fixed Gaussian spike train. The mean of the Gaussian spike train is picked from a flat random distribution between 10 Hz and 80 Hz. Encoders values are randomly picked between two



(a) (b)

Figure 8.2: 8.2a. Tuning functions for all neurons. 8.2b. Encoding and decoding functions. $f(x) = x$ $RMSE = 0.0155$, $f(x) = x^2$ $RMSE = 0.0287$, $f(x) = x^3$ $RMSE = 0.0187$.

alternatives (+1,−1), this gives the different directions of the tuning curves in Fig. 8.2a. Example of encoded and decoded mathematical functions are shown in Fig. 8.2b. The protocol of this experiment is the stimulation of all neurons with a swept Poisson spike train from 1 Hz, to 200 Hz in 25 steps of 200 ms. The optimal linear decoders, d_i, used in Fig. 8.2b, are estimated by minimizing the expression $\langle (x - \hat{x})^2 \rangle_x$ with the least square method under some expected mean zero, independent Gaussian noise.

8.2.3 Real-time computations of mathematical functions

Mathematical computation can be performed by pools of neurons in which the encoded representations are defined by

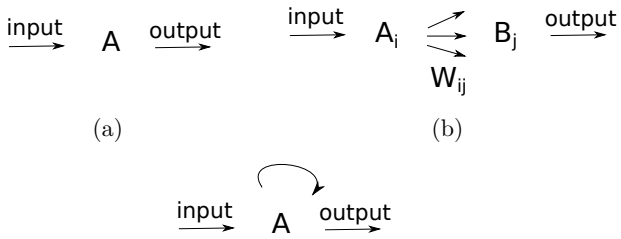
$$a_i = G_i[\alpha_i \langle x \cdot e_i \rangle + J_i^{bias}], b_j = G_j[\alpha_j \langle y \cdot e_j \rangle + J_j^{bias}], \tag{8.6}$$

and the respective representational decodings are:

$$\hat{x} = \sum_i a_i d_i, \hat{y} = \sum_i b_j d_j. \tag{8.7}$$

Note that it is possible to find optimal decoders for arbitrary nonlinear functions of x using this same technique. We denote these as $d_i^{f(x)}$. It is thus possible to compute the desired mathematical computation, by substituting estimates of the desired function into b such that $y = f(x) \approx \hat{f}(x)$ we obtain

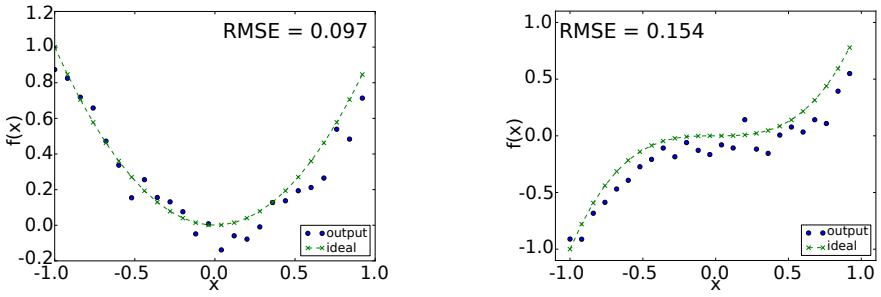
$$b_j = G_i \left[\sum_i \omega_{ji} a_i + J_i^{bias} \right], \tag{8.8}$$



(a) (b)

(c)

Figure 8.3: Schematic diagrams. 8.3a. Encoding with neurons. 8.3b. Transformations. 8.3c. Integrator.



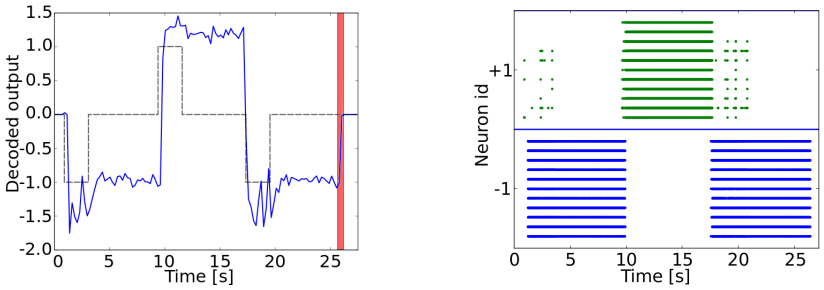
(a) (b)

Figure 8.4: Computing mathematical functions with neurons. 8.4a. computed function $f(x) = x^2$, $RMSE = 0.097$. 8.4b. computed function $f(x) = x^3$, $RMSE = 0.154$.

in which the weights matrix is $\omega_{ji} = a_j e_j d_i^{f(x)}$. The schematic representations in terms of pool of neurons is shown in Figure 8.3b. We used 28 neurons for each population (A, B). All to all connections are realized using the mapper board. We achieved computation as shown in Figure 8.2. Neurons in population A are excited with a ramping Poisson spike train from 1 Hz to 200 Hz in step of 200 ms. This is equivalent to the input range [−1,1]. Population A spiking activity is in real-time directed to population B whose output is the desired computed mathematical function.

8.2.4 Working memory as a dynamical system

We implemented a stable dynamical system by introducing recurrent connections in the network. We realized the neural correlates of working memory, this means that the network dynamics is capable of storing input values through self-sustained activity. To achieve memory states, we implemented the dynamics of an integrator, (see Fig. 8.3c).



(a) (b)

Figure 8.5: Dynamical system: integrator. 8.5a. Decoded activity from the pool of neurons recurrently connected. The dotted line represents input to the network, note that when input is removed the network stores its integrated value in an attractor state. The red stripe, $t = 26s$, defines an inhibitory input stimulus sent via the computer. 8.5b. Raster plot of network activity, the two colors (blue, green) represent neurons with different encoders (+1 or -1).

The third principle of NEF describes the relationship between standard control theory and neural dynamics, in this mapping an integrator is described by the transformation matrices in which $A' = 1$ and $B' = \tau_{PSC}$. Moreover, if we assume an exponential postsynaptic current ($h_{PSC}(t) = \frac{1}{\tau} e^{\frac{-t}{\tau}}$) and a linear time-invariant system, recurrent weights can be computed from the dynamical matrices of the system. We used a pool of 22 neurons recurrently connected. Encoder values where randomly picked between two alternatives (+1,−1). In Fig. 8.5 we show the activity over time of the integrator. Figure 8.5a shows in dotted line the input generated by the computer and in continuous line the decoded output value. At first, we excite the neural pool with an input directed to the neuron encoding for −1, and at the removal of the input, the network successfully store the −1 value in a reverberant state of activity. At $t = 10\,\mathrm{s}$, we excite the the network with a +1 stimulus and the network correctly stores it. After storing one more −1 value (at $t = 16\,\mathrm{s}$), we kill the activity of the network at $t = 26\,\mathrm{s}$ with an inhibitory input stimulus. Figure 8.5b shows the raster plot of the run.

8.2.5 Discussion: Spike-based distributed analog computing in heterogeneous neuromorphic systems

This work addresses the challenge of obtaining distributed and programmable computation with noisy and heterogeneous analog circuits in a network of spiking neurons. We demonstrated that arbitrary computation in a neuromorphic multi-neuron VLSI chip can be achieved using the NEF's principles. The NEF framework represents a robust method for computing connection weights that takes into account neurons with a wide range of

different transfer characteristic. This requirement makes the framework appealing to analog computation, as the effect of device mismatch is compatible with the diversity requirements, and allows for very compact neuron designs. On the other hand, the method requires precise synaptic weights. To compensate for the negative effects of mismatch in the neuron and synapse circuits, we exploited the availability of programmable SRAM cells, while keeping the synapse and neuron circuits compact. As demonstrated, we showed reliable computation of functions across different pools of neurons. Additionally, we constructed robust dynamic attractor states by introducing recurrent connections in the network. Such stable dynamical systems are fundamental for building complex neural systems, and developing brain inspired spike-based computing systems. The attractor experiment demonstrates how it is possible to implement real-time state dependent computation and reliable memory storage using sets of 22 slow and imprecise silicon neurons. The implementation of plausible neural collective dynamics in neuromorphic substrates is an important step also for future nano-technologies that are likely to rely on mismatched and unreliable components.

CHAPTER 9

A neuromorphic event-based neural recording system for smart Brain-Machine-Interfaces

F. Corradi, and G. Indiveri

Biomedical Circuits and Systems, IEEE Transactions on, (submitted), 2015

9.1 Introduction

Neural recording systems are a central component of Brain-Machine Interfaces (BMIs). In most of these systems the emphasis is on faithful reproduction and transmission of the recorded signal to remote systems for further processing or data analysis. Here we follow an alternative approach: we propose a neural recording system that can be directly interfaced locally to neuromorphic spiking neural processing circuits for compressing the large amounts of data recorded, carrying out signal processing and neural computation to extract relevant information, and transmitting only the low-bandwidth outcome of the processing to remote computing or actuating modules. The fabricated system includes a low-noise amplifier, a delta-modulator analog-to-digital converter, and a low-power band-pass filter. The bio-amplifier has a programmable gain of 45-54 dB, with a Root Mean Squared (RMS) input-referred noise level of 2.1 μV, and consumes 90 μW. The band-pass filter and delta-modulator circuits include asynchronous handshaking interface logic compatible with event-based communication protocols. We describe the properties of the neural recording circuits, validating them with experimental measurements, and present system-level application examples, by interfacing these circuits to a reconfigurable neuromorphic processor comprising an array of spiking neurons with plastic and dynamic synapses. The pool of neurons within the neuromorphic processor was configured to

implement a recurrent neural network, and to process the events generated by the neural recording system in order to carry out pattern recognition.

9.1.1 Motivations

Neural recording systems are typically concerned with the acquisition of signals from the nervous tissue and with the transmission of these signals off-chip for further processing and analysis. As the signals being recorded are typically very small and noisy, most of the research and development efforts have been directed toward the construction of very low-noise, low-power, and high-gain amplifiers [Borghi et al., 2007; Harrison, 2008]. To transmit the amplified signals off-chip, much research has also been dedicated to the design of wireless data links [Charles, 2007; Mandal and Sarpeshkar, 2007; Nurmikko et al., 2010]. Thanks to these efforts, there has been tremendous progress in the development of BMIs that make use of these implanted microelectronic systems to reproduce as faithfully as possible the signals recorded from the neural tissue and to transmit as much of this information as possible to off-line processing stages [Aziz et al., 2009; Wattanapanitch and Sarpeshkar, 2011; Wise et al., 2008]. The off-line computers and signal processing stages are then typically used to process the vasts amount of data being transmitted, for extracting information from interacting populations of neural cells, and for detecting action potentials, sorting them and labeling them according to the potentially multiple neurons that produced them. These systems represent extremely important tools for aiding fundamental research in neuroscience; however as the amount of electrodes for simultaneous recordings in the nervous tissue scales to very large numbers [Lopez et al., 2014; Stevenson and Kording, 2011], the energy and bandwidth required to transmit the raw data to off-chip processing stages increases to levels that are prohibitive for systems that are expected to be chronically implanted, close to the living tissue. To build neural prosthetic devices that can extract information from very large numbers of neurons and decode them *in-situ* without transmitting this information off-line, it is necessary to develop additional ultra low-power processing stages that can be interfaced to the low-noise neural signal amplifiers and integrated on the same die. In these application scenarios, it becomes important to maximize the information extracted from the raw signals, without necessarily detecting and sorting the action potentials produced by the neurons sensed by the electrodes [Zhang et al., 2015]. BMIs typically make use of these implanted microelectronic systems to reproduce as faithfully as possible the signals recorded *in-situ* from the neural tissue and to transmit as much of this information as possible to off-line processing stages [Aziz et al., 2009; Lopez et al., 2014; Wattanapanitch and Sarpeshkar, 2011]. Bulky and power-hungry computers and signal processing systems are then typically used to process the vast amount of data being

transmitted, for extracting information from interacting populations of neural cells, decoding the neural signals, interpreting planning activities and motor commands, and eventually executing prosthetic control commands. While these systems are extremely important for fundamental research in neuroscience and specific applications that do not require compact mobile solutions, it is also important to develop alternative solutions that might not be optimal for faithfully reproducing the signals recorded, but are compact, low-power, real-time, and implantable for constructing neural signal processing systems that can carry out some of the tasks required for prosthetic control directly by the electrodes. This is analogous to the domain of vision, with the duality between high-resolution camera sensors that are optimized to reproduce images, and neuromorphic vision sensors [Liu and Delbruck, 2010] that are designed to process visual signals quickly and efficiently. In this paper we present a neural recording and processing system which converts the recorded bio-signals into asynchronous digital events and sends them to a low-power spiking neural network endowed with adaptive and learning abilities for decoding and classifying them on-line. We describe the features of the neural recording and signal conditioning circuits, and present demonstrations of *in-situ* signal processing using the neuromorphic architecture. The neural recording part of the system comprises a set of circuits that record, amplify, filter, and convert the bio-signals into digital asynchronous streams of pulses (see Fig. 9.1), which are then encoded using the AER [Deiss et al., 1998]. This representation is commonly used in neuromorphic systems to implement an asynchronous communication protocol that routes and maps address-events from multiple source nodes to multiple destinations. Typically the sources of AEs are pixels or nodes of neuromorphic sensory systems [Liu and Delbruck, 2010], or silicon neurons in multi-neuron chips [Indiveri et al., 2011]. Neuromorphic spiking neural networks can then be used to process these AEs using hardware emulations of synapses with on-line learning abilities [Chicca et al., 2014], for implementing compact and low power cognitive systems that learn and adapt to the changes in the statistics of the signals being processed [Neftci et al., 2013]. The goal of this work is to develop a set of circuits that can convert neural signals and act as sources of events, very much like vision sensor pixels or silicon neurons in AER neuromorphic systems, and to demonstrate learning and adaptive abilities of the neuromorphic architecture connected to them for reconstructing and classifying the recorded neural signals. The combined event-based neural recording circuits and neuromorphic learning architecture represent self-contained "smart" BMI able to produce relevant low-bandwidth control signals for prosthetic actuators, without requiring power-demanding wireless transmission of raw neural data to off-chip processing units. To record neural-, and in general bio-, signals, a standard Low-Noise Amplifier (LNA), analogous to the one originally proposed in [Harrison and Charles, 2003]. The output of

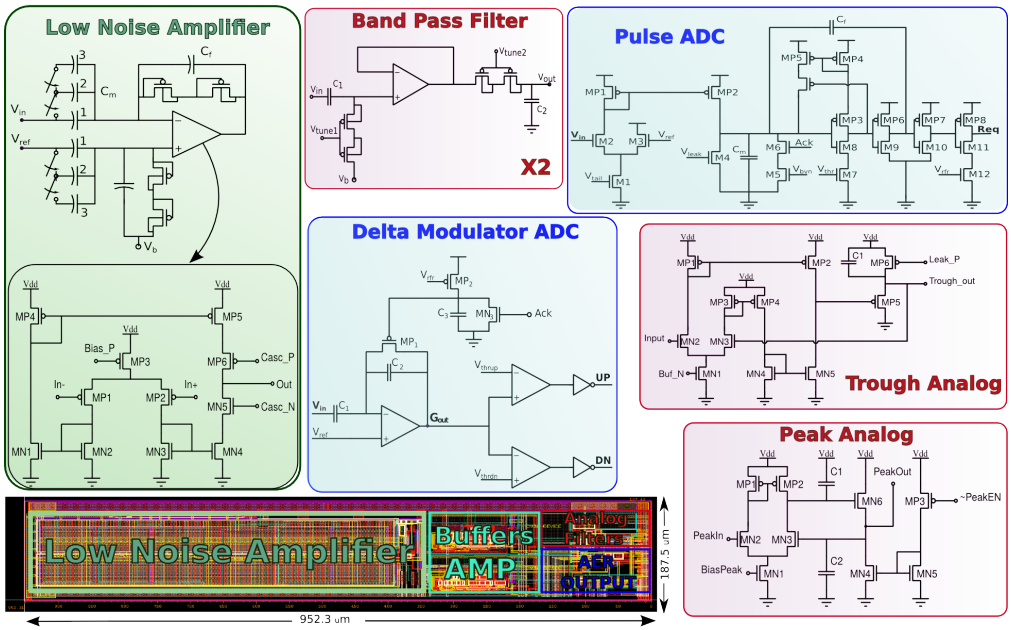


Figure 9.1: Neural recording system block diagram. Bio-signals are amplified by the low-noise amplifier circuit. The output voltage of this circuit is then sent to five blocks: a) a cascade of two band-pass filters, b) an AER A/D Delta modulator, c) an analog peak detector circuit, d) an analog trough detector circuit, and e) an AER threshold-crossing spike detector circuit.

the LNA is sent to an Analog-to-Digital Delta Modulator analogous to the asynchronous analog-to-digital conversion system proposed in [Tang et al., 2013], and similar to the event-generating circuits used in neuromorphic vision sensors [Lichtsteiner et al., 2008], but extended with AER asynchronous communication handshaking circuits for producing the desired AEs. In addition to converting the raw amplified bio-signals into AEs, we connect the amplifier output also to a band-pass filter, analogous to the one proposed in [Wattanapanitch and Sarpeshkar, 2011], also extended to produce AER output.

This chapter is organized as follow: in the next section we describe the neural recording and processing system circuits, together with the analog to digital converters. In Section III we provide experimental measurements from the neuromorphic neural architecture that characterize the response properties of all the neural recording circuits and that demonstrate the ability of the neuromorphic architecture to classify the signals produced by the neural recording circuits. In the Discussion section we discuss about other possible neural models, alternative to the one used for improving the classification of sequences of syllables, and about alternative spiking neural network learning frameworks that can be used to decode the address-events produced by the system. In the Conclusions we

summarize the achievements made and present a brief outlook for future work.

9.2 System and circuits

The overall architecture of the neural recording system integrated with the neuromorphic spiking neural network is shown in Fig. 9.1. The amplifier designed to record neural-, and in general bio-, signals, is a standard LNA, analogous to the one originally proposed in [Harrison and Charles, 2003], but extended with an Analog-to-Digital Delta Modulator similar to the one proposed in [Tang et al., 2013], with event-generating circuits similar to those used in neuromorphic vision sensors [Lichtsteiner et al., 2008], and with AER asynchronous communication handshaking circuits for producing the desired AEs. In parallel, the analog output of the LNA is sent to other five main blocks: a band-pass filter with pulse ADC output, an ADC delta modulator, two analog "peak" and "trough" filter circuits [Horiuchi et al., 2004, 2007], and a basic threshold-crossing spike detector circuit, to investigate potential spike-sorting capabilities of the system. All circuits were fabricated using a standard $0.18\,\mu m$ 1-poly 6-metal CMOS technology. The complete layout of the system occupies an area of $0.178\,mm^2$. The low noise amplifier was designed for amplifying bio-signals in the μV range, while rejecting the large DC component that is present at the electrode-tissue interface. In fact, action potentials typically have amplitudes that range from $5-500\,\mu V$, depending on the distance to the electrode and on cell's size. The bandwidth of the action potential signals is in the range of $100\,Hz-3\,kHz$, whereas the Local Field Potential (LFP)s are in the frequency range $0.5-300\,Hz$.

9.2.1 Low noise amplifier

The low noise amplifier was designed for amplifying bio-signals in the μV range, while rejecting the large DC component that is present at the electrode-tissue interface. Indeed, action potentials typically have amplitudes that range from $5-500\,\mu V$, depending on the distance to the electrode and on cell's size. The bandwidth of the action potential signals is in the range of $100\,Hz-3\,kHz$, whereas the LFPs are in the frequency range $0.5-300\,Hz$. The circuit schematic of the low noise amplifier is shown in the first column of Fig. 9.1. The amplifier is a capacitive feedback circuit mid-band gain $A_M = -C_f/C_m$, micro-volt range input-referred-noise, and bandwidth $\approx g_m/(A_M C_L)$, where C_L represents the load capacitance The capacitive feedback is formed by the capacitors C_f, and C_m. C_f is in parallel to the pseudo-resistor Metal Oxide Semiconductor Field Effect Transistors (MOSFETs) and connects the inverting input to the output of the amplifier. In this design, an impedance element is used to constraint the gain by a proportional bandwidth reduction. The feedback loop controls the output signal while providing a

Table 9.1: Low Noise Amplifier operating points

Devices	W/L [μm]	I_d
MP3	8/4	1.3 (μA)
MP1	720/4	125 (nA)
MP2	720/4	1.25 (μA)
MN2	2/16	125 (nA)
MN1	2/16	125 (nA)
MN3	2/16	125 (μA)
MN4	2/16	125 (nA)
MN5	8/18	125 (nA)
MP6	6/16	125 (nA)
MP5	6.4/6	125 (nA)
MP4	6.4/6	125 (nA)
MP3	8/4	125 (nA)

controllable linear response. We used Metal-insulator-metal Capacitor (CMIM) capacitors of values $C_f = 35\,fF, C_{m1,m2,m3} = 15\,pF$. that are placed on top of the active area. The input switches in Fig. 9.1 represent transmission-gate switches that are used to vary the total C_m value, resulting in a programmable gain. The differential inputs are capacitively coupled to reject the DC potential differences between electrode V_{in} and reference signal V_{ref}. The main source of noise in the system is thermal noise. The MOSFET pseudo-resistors, in parallel to the capacitor C_f create the low-pass filter with a cut-off frequency at a few Hz. This design has been introduced in [Harrison, 2008; Wattanapanitch and Sarpeshkar, 2011], and it represents a good compromise between performances, power and silicon area. The very low-frequency pole is achieved thanks to the fact that the MOSFET pseudo-resistor elements (M_{p1}, M_{p2}) have a much higher impedance than weak-inversion transistors, i.e., the input transistors of the amplifier. To achieve low input-referred-noise it is therefore important that the input transistors work in weak inversion with microamp current levels: this can be achieved thanks to the use of very wide input transistors, as can be seen in Fig. 9.1.

The transistor sizes for all transistors that are part of the LNA, and the DC operating point currents are shown in table 9.1.

9.2.2 Band-pass filter circuit with AER output

The band-pass filter has a low cut-off frequency that depends on the properties of a Metal Oxide Semiconductor (MOS)–bipolar pseudo-resistor and the capacitor used, as described in [Wattanapanitch and Sarpeshkar, 2011]. There is however also a bias parameter that can be used to tune low-pass and high-pass cutoff frequencies. The value of this bias

regulates the band of interest, for action-potentials or LFPs. The pulse-output circuit converts the filtered signal into asynchronous spikes. The circuit is shown in Fig. 9.1. The filtered signal is converted into a current via a differential pair operated in the weak-inversion, or subthreshold, domain and controlled by the V_{ref} and the V_{tail} bias voltages. This current is injected into the capacitor C_m. Constant positive offsets can be removed thanks to a leak term, added via the transistor M_4 and set by its bias voltage V_{leak}. As the voltage across the capacitor C_m slowly reaches the switching threshold of the first inverter $M_7 - MP_5$, a positive feedback loop quickly switches it, minimizing the switching current, and triggers a cascade of inverters which eventually generate the AER handshaking Req request signal. The circuit is reset by the Ack acknowledgment signal, produced by the AER asynchronous four-phase handshaking circuits (not shown).

9.2.3 The asynchronous Delta Modulation A/D converter

The schematic diagram of the Delta Modulation A/D converter circuit is shown in Fig. 9.1. It consists of an input operational transconductance amplifier with a capacitive-divider gain stage, two comparators, and additional analog/digital circuits to manage the AER handshaking interface. Functionally, this circuit is equivalent to the one used in the DVS [Lichtsteiner et al., 2008]: it is a self-timed clock-less circuit which produces two types of digital pulses (UP or DN) if the input signal V_{in} changes by a fixed positive or negative amount respectively. The output pulses UP and DN of the circuit correspond to handshaking request signals for the AER interface. As either of these pulses are produced, the AER receiving circuits will respond with an acknowledgment signal ACK which resets the comparator output to the reference voltage V_{ref}, by shorting the amplifier output to its negative input terminal via the MP_1 p-FET. This reset state will be held for a period that is determined by the values of the C_3 capacitor and MP_2 leak current. This is essentially a "refractory period" which can be used to limit the maximum rate of AEs produced by the circuit (e.g., to control bandwidth usage).

9.2.4 The AER communication scheme

A handshaking mechanism ensures that all the digital UP and DN events generated at the sender side arrive at the receiver. These signals are encoded using a Bundled Data (BD) representation, in which the address of the channel is conveyed as a parallel word, together with two additional ACK and REQ signals that are required for the handshaking control sequence. In this system time represents itself, this means that AE are communicated only when an event is generated and without the use of a clock in the device itself. In the case of a multi-sender situation, an arbitration block ensures that addresses do not collide, but are transmitted on the bus in sequence. The throughput of such AER systems

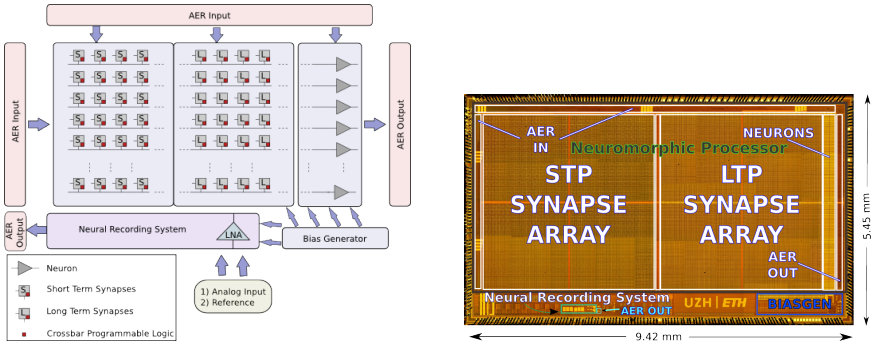
is usually in the order of 5 megaevents/s [Mostafa et al., 2013]. To manage the ADCs output signals and to route the AEs produced, we used an off-chip commercial FPGA device (spartan$-VI$). This device represents an optimal development platform which was used mainly for data logging and AE routing.

9.2.5 The analog filters: peak detector, trough detector and level crossing

The analog filters include a peak detector circuit, a trough detector circuit and a comparator that acts as a spike detector circuit. This spike detector circuit is used to trigger the measurement of the spike amplitude. The peak and trough circuits are asymmetric voltage-followers in which the amplifier gain and the output offset voltage effect the output of these circuits. They are used to measure the peak amplitude of a spike and the deepness of the hyper-polarization of action potential signals. These features have been demonstrated to be extremely useful in spike-sorting applications [Barsakcioglu et al., 2014; Horiuchi et al., 2007]. In the peak detector circuit, the MOSFET $MN6$ acts as a switching element, while the capacitor $C1$ is the charging element. The circuit is triggered upon the arrival of a digital $PeakEN$ signal produced by the comparator circuit (see the Spike Detector in Fig. 9.1).

9.2.6 The reconfigurable neuromorphic processor

We used a reconfigurable spiking neural network processor [Qiao et al., 2015] for carrying out neural processing tasks. The NP of Fig. 9.2a contains 256 adaptive exponential integrate-and-fire neurons implemented in a mixed signal analog/digital design. There are $128\,k$ synapses, of which $64\,k$ implement a LTP Hebbian-like plasticity rules [Brader et al., 2007; Mitra et al., 2009], and $64\,k$ synapses have two possible programmable weights resolution, in addition to the possibility to configure them as either excitatory or inhibitory. The synaptic matices allows on-chip connectivity thanks to a crossbar structure. In principle all-to-all connections are possible, depending on the programmable logic state. The LTP synapses comprise pre-synaptic spike-based learning circuits with bi-stable synaptic weights [Mostafa et al., 2014]. Additional circuits are also instantiated next to the neurons array. These additional circuits are needed in order to implement the spike-based weight update algorithm [Brader et al., 2007], and they represents the calcium concentration at the post-synaptic side. The internal dynamics of the synapse is analog but the state of it is binary, this removes the need of storing analog variables on long-time scales and simplify the circuit implementation. We refer the reader to Section 3.3.1 for a description of the mathematical model of the learning rule, as well as circuits description. Both the neural network architecture and the parameters of the neuromorphic core are



(a) (b)

Figure 9.2: (a) Schematic diagram of the die comprising both neural amplifier and neuromorphic architecture. The neuromorphic processor contains 256 neurons, and two crossbar matrices of synapses, each containing 64k synapses. The STP synapse array and the LTP synapse array. Network connections are programmable thanks to the crossbar programmable logic. The STP synapses have fixed programmable weights, while the LTP synapses implement an Hebbian-like plasticity rule. A bias generator is used to configure the analog parameters of the two systems. (b) Die micro graph. The neural recording system and the neuromorphic processor are located in two distinct areas of the same VLSI chip. The neural recording system occupies a small portion (i.e. $\approx 0.178\,mm^2$) of the die (entire die area is $\sim 51.4mm^2$). The two blocks communicate thanks to the AER protocol.

fully programmable via a high-level Python framework [Stefanini et al., 2014]. The combination of reconfigurable hardware with the Python-based configuration framework supports the real-time emulation of a wide range of spiking neural network architectures. This setup enable us to easily process recorded neural signal using real-time hardware implementations of spiking neural networks, and to design intelligent systems capable of performing spike-pattern recognition by exploring different neural network architectures.

9.2.7 The on-chip learning rule

The LTP synapse array in the neuromorphic architecture implements a spike-based Hebbian-like learning rule [Brader et al., 2007]. The weight update rule depends on the timing of pre-synaptic spike, on the the state of the post-synaptic neuron's membrane potential, and on the recent firing history of the post-synaptic neuron. While the circuits that implement this algorithm are deterministic, there is a source of stochasticity in the pre and post-synaptic spike trains, that is used to avoid updating all the synapses in the same way. A mathematical description of the weight update rule and its circuit description can be found in Chapter 5.

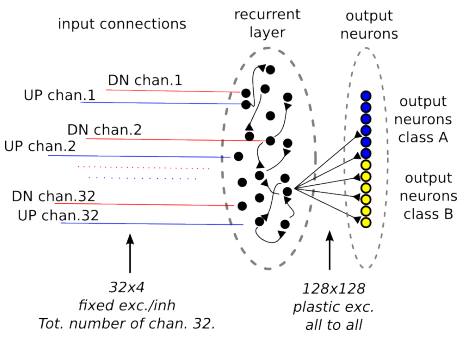


Figure 9.3: On-chip network architecture. UP/DN channels represent the spiking output from the ADC delta modulator.

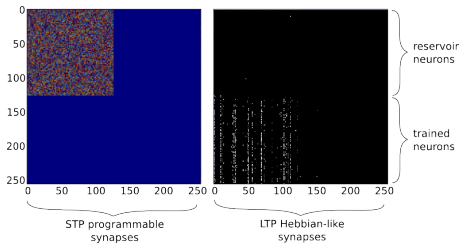
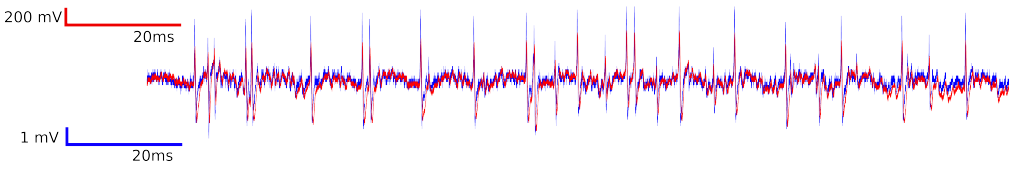


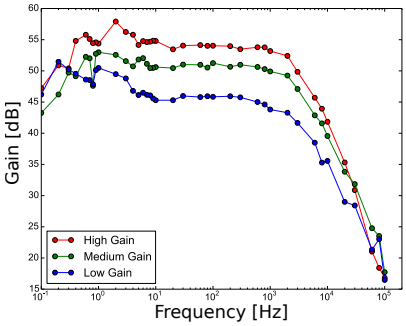
Figure 9.4: Neuromorphic chip configuration. Right side: STP fixed weight programmable synapse matrix. Left side: LTP learning synapse matrix.

9.2.8 The neuromorphic classifier's architecture

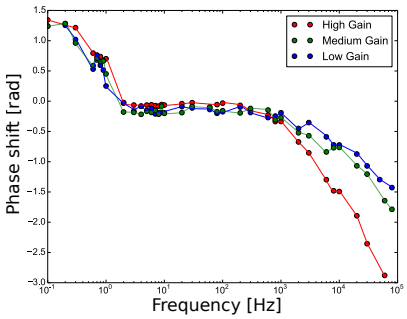
The neural core processor has been configured to implement a network of two layers, as shown in Fig. 9.3. The first layer of the network is composed of a recurrent pool of 128 neurons that acts as a spiking reservoir: it receives spike-event inputs from the neural recording system, and it exploits the analog dynamics of the neurons and synapses to enhance the temporal properties of input patterns. All the neurons in the recurrent pool are connected with all-to-all projections to a second read-out layer of 128 neurons. The on-chip reservoir layer remains unchanged during training, while the connections between the first and second layer are plastic: they change their weights depending on the activity of pre and post-synaptic neurons, as described in Section 9.2.7. In the initial configuration these synaptic contacts are reset to their low state. Figure 9.4 shows the configuration of the synaptic structure of the neuromorphic processor that implements the network shown in Fig. 9.3. The recurrent pool of neurons is implemented in the first 128 neurons that are recurrently connected via the synapses of the STP array. Every dot in the Fig. 9.4 represents a synaptic contact. Colors at the left side of Fig. 9.4 encode the type of connection. These connections are distributed among two possible weights and among two possible types: excitatory and inhibitory. The connections are initialized at random, and every neuron in the recurrent pool receives in average 120 connections of which 85 are excitatory and 35 are inhibitory. The right side of Fig. 9.4 shows the LTP synapse matrix. Synapses in the high state are colored white. This figure evidences the state of synapses after few transitions, to show the quadrant in which connections are made.



(a) LNA Input and Output



(b) LNA Gain



(c) LNA Phase Margin

Figure 9.5: a) Input and output of the Low-Noise Amplifier stage, the signal is a neural recording from a cricket (*Mecopoda Elongata*). Amplifiers is set to the minimum programmable gain (45 dB). b) Measured gain for three different values of input capacitors. c) Measured phase shift.

9.3 Results

9.3.1 Low noise amplifier measurements

To characterize the response properties of the LNA to signals with magnitude comparable to that of physiological ones, we applied an input sine wave produced by a Hewlett Packard 33120*A* Function Generator, setting the amplitude such that after a 100:1 resistive voltage divider the signal was 0.61 mV (the minimum amplitude range measurable by our oscilloscope). The voltage divider was a standard 2-resistor circuit, consisting of a 24.9 $k\Omega$ resistor and a 249 Ω resistor. Measurements were taken for three different gain configurations, set externally with jumpers to activate or deactivate the input capacitors shown in Fig. 9.1. The transfer function for the three gain settings is show in Fig. 9.5b. The highest gain setting gives a maximum of 58 dB, and a flat-band gain of 54 dB between 10 Hz and 2.5 kHz. The circuit still has strong amplification in the low frequency range, but the input has noticeable deformations for frequencies below 3 Hz. As shown in Fig. 9.5c, the circuit provides very little phase shift in the 2 Hz-200 Hz range, and it is smaller than $-\pi/2$ rads until above 7 kHz.The gain seems to affect the phase of the output signal at high-frequency ends (see Fig. 9.5c). We also tested the LNA with prerecorded signals from the auditory system of a cricket (Mecopoda Elongata), in which

a single electrode was used (data courtesy of Prof. Manfred Hartbauer). The recordings were played with a standard computer audio card and attenuated to scale the signals to physiological levels. In Fig. 9.5a we show the result of this experiment. Since the input signal is in the millivolt range, the amplifier has been set to the minimum gain. This produces an output that is in the range of $200mv$, with no visible deformations on the amplified signal. The effective NEF achieved (see table 9.2) is not impressive if compared to state-of-the art amplifiers for neural recordings as in [Chen et al., 2014; Han et al., 2013; Kmon and Grybos, 2013]. This is partially due to the limited bandwidth of our amplifier in comparison to state-of-the art implementations that currently achieve effective NEF 4,5 time smaller with similar input-referred-noise values.

9.3.2 The A/D asynchronous delta modulator

The output measured from the delta modulator circuit in response to a sine-wave input is shown in Fig. 9.6a. The top trace shows the output of the transconductance amplifier G_{out}, together with a reconstruction of the signal. When G_{out} exceeds one of the two thresholds V_{thrup} or V_{thrdn}, the UP pulse (third trace from the top) or DN pulse (bottom trace) are produced, and G_{out} is reset to $V_{ref} = V_{dd}/2 = 900\,mV$. The threshold voltages of the delta modulator are set to $V_{thrup} = V_{dd}/2 + V_{dd}/10$, $V_{thrdn} = V_{dd}/2 - V_{dd}/10$

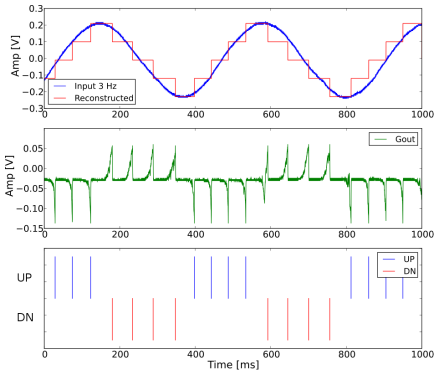
Decoding the UP and DN events

The reconstruction of the input signal from the UP and DN AEs is carried by the execution of the following algorithm: The parameters δ_{UP} and δ_{DN} represent the incremental step

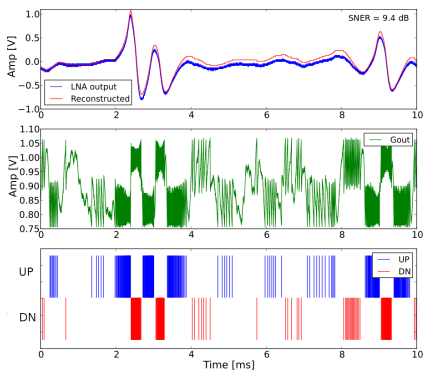
Result: *reconstructed*
while *incoming events* **do**
  $reconstructed(t) = reconstructed(t-1)$;
  if *channel events* == *UP* **then**
    $reconstructed(t) = reconstructed(t-1) + \delta_{UP}$;
  end
  if *channel events* == *DN* **then**
    $reconstructed(t) = reconstructed(t-1) - \delta_{DN}$;
  end
end

Algorithm 1: Signal reconstruction algorithm

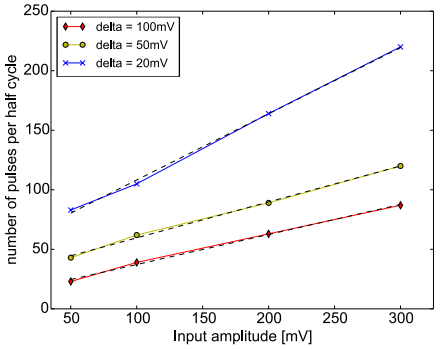
caused by a single AE. For the scope of this demonstration, we calibrated the ADC delta modulator to output the same amount of spikes with a sinusoidal input, as in Fig. 9.6a. Figure 9.6b shows a reconstruction of an action potential measurement. The top plot shows the action-potential signal after amplification with the LNA, superimposed to the



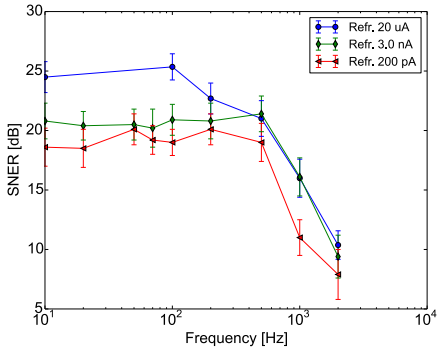
(a) ADC Delta Modulator



(b) Action potential signal reconstruction



(c) ADC Delta Modulator linearity



(d) ADC Delta Modulator refractory period

Figure 9.6: a) ADC delta modulator thresholds, calibration, and signal reconstruction. b) action-potential signal reconstruction with refractory period set to 200 pA. d) Signal to noise error ratio for different input sine wave frequencies. The amplitude of the input signal is set to 50 mV. c) Linearity of the converter over different input amplitudes. The frequency of the sine wave is set to 100 Hz.

reconstruction of the signal from the UP and DN events; the middle column shows the node V_{gout} of the asynchronous delta modulator that is constantly being compared with the thresholds; the bottom trace shows the UP and DN events. The signal to noise ratio between the two signals is $SNR = 9.4dB$. In Fig. 9.6d we show the effect of the refractory period on the signal to noise error ratio. When the system is in the refractory period the input signal is discarded and for this fact the error accumulates. In Fig. 9.6c we present a measure of the linearity of the ADC as a function of the amplitude of the input signal and of the delta step. The input signal used is a sine wave of 100 Hz. Changes in the δ step are reversely proportional to the total number of pulses, for a given input signal amplitude.

The programmable band-pass filter has been tested with a square wave as input. It was tuned by controlling the low-pass and high-pass parameters to record either LFPs or neural spikes, as evidenced in Fig. 9.7a. Figure 9.7a shows the response of the circuit for the two bias settings. Figure 9.7b shows the output of the circuit for a high-pass filtered square wave: the lower green plot is the output of the pulse circuit, which shows a linear increase in frequency for lower amplitudes and a decrease of frequency for higher amplitude.

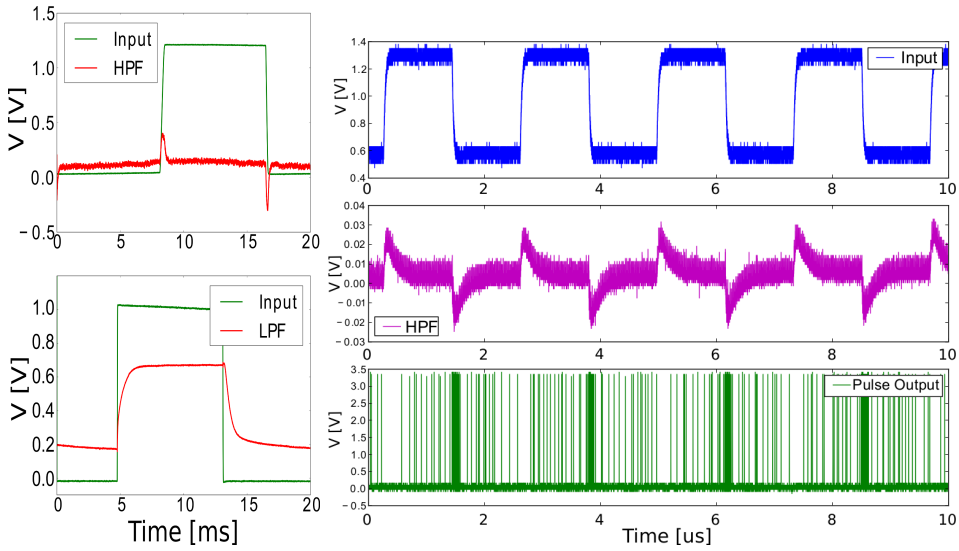
9.3.3 Pulse ADC characterization

Figure 9.8 shows the input and output of the ADC pulse circuit for a linear sweep in the input voltage, in the range $0.5 < V_{in} < 0.8$ V. The line plot in the figure shows the reconstruction of the input signal. Figure 9.8b shows a measurement output frequency of the pulse ADC as a function of the input current. This curve is linear in the mid range of input currents, i.e., $10^{-10} < I_{in} < 10^{-6}$ A; while it has a clear non-linear drop in frequency for input currents below $I_{in} < 10^{-10}$ A. This drop is caused by the non-linearities of the input DPI, used to convert V_{in} to the current signal I_d of transistor $MP2$ (see Fig. 9.1 Pulse ADC). Signal reconstruction is carried by averaging the mean rate activity of the asynchronous digital output events. To obtain an absolute estimate of the input current, one has to interpolate the mean rate frequency with the characteristic visible in Fig. 9.8b.

9.3.4 Binary classification of neuro-biological recordings

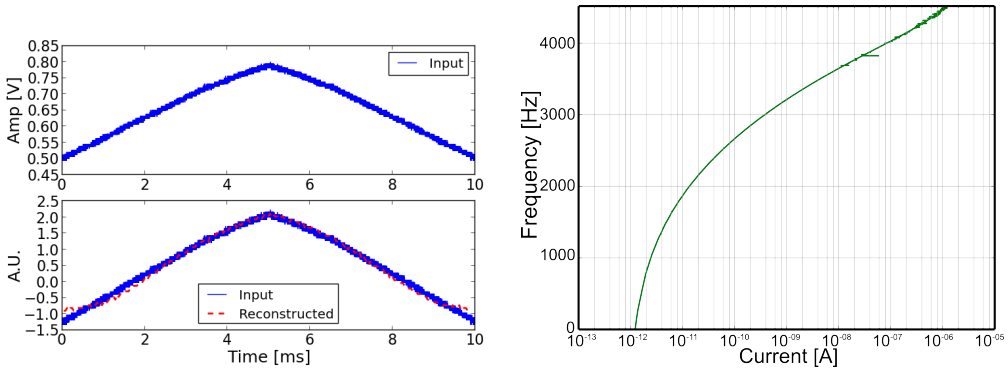
Neural recording data and classification task

In this experiment we demonstrate the capabilities of the full system composed of the neural recording system and the neuromorphic processor. In this demonstration we use real neuro-biological recordings from Zebra Finches, a passerine bird from Central



(a) (b)

Figure 9.7: (a). Band-pass filter measurements. (b). Pulse-output circuit measurements. Figure from [Corradi et al., 2014a].



(a) (b)

Figure 9.8: a) ADC Pulse recontruction. b) Output frequency versus input current.

Table 9.2: Neural Recording System characteristics

Technology (CMOS)	$0.18\,\mu$m 1P6M
Supply Voltage	1.8 V
Total area	$0.178\,mm^2$
LNA power	$90\,\mu$W
LNA RMS input-referred-noise	$2.1\,\mu$V
A/D Delta Modulator power	100 Hz $55\,\mu$W
Bandwidth	$3\,Hz - 2.5\,kHz$
Gain (programmable)	54/50/45 dB
Effective NEF @ 45 dB	11

Australia. The data were taken in an anesthetized bird, in the lab of Prof. Hahnloser, and were kindly provided by Jenie Ondracek. The data consist of signals measured by four electrodes of auditory-forebrain neurons while natural sounds were being used as auditory stimuli. We focused on classifying two classes of auditory stimuli that have similar average energies, but different temporal structures: the bird's own song (BOS) (see Fig. 9.10a), and the reversed version of the bird's own song (REV) (see Fig. 9.10b). To this end, we grouped multiple channel recordings for the same class of auditory stimuli (BOS and REV respectively). In Fig. 9.9a we show such grouping for 32 recordings in response to the BOS stimulus. The onset of the stimulus is aligned with the onset of the recordings. Figure 9.9b shows the output of the delta modulator for 40 ms. The reconstructed trace is shown in Fig. 9.9c. Even though the reconstruction does not faithfully reproduce the original recordings, all action potentials in the traces are preserved. To simulate a multi-electrode recording device with many 32 parallel channels, we grouped 32 electrode neural recording signals.

Traning the neuromorphic classifier

Training the binary classifier is done by presenting multiple trials comprising signals obtained from grouped neural recordings, jointly with "teacher" signals generated externally. These teacher signals are Poisson spike trains that have a mean frequency of 25 Hz (teacher-false signal) or 150 Hz (teacher-true signal). These synthetic Poisson spike trains are directed to 4 AER virtual synapses which drive the neurons in the perceptron layer. The teacher-true signal drives the perceptron neurons such that they fire at a high rate (of approximately 100 Hz) when the true-class input stimulus is present, while the teacher-false signal drives neurons to fire at low frequency (about 5 Hz) when the false-class stimulus is applied. In this way, the synapses that connect the reservoir pool

neurons to the perceptron layer neurons, driven by a teacher-true signal will tend to potentiate, making transitions to the high binary state; while synapses that connect reservoir neurons to the perceptron layer neurons driven by the teaching-false signal will tend to make transitions to the low binary state. At the beginning of the training session, all learning synapses are set to the low binary state. The training procedure of the classifier consists of presenting three repetitions of the same grouped recording trial, together with the appropriate teacher signal. In this condition a random subset of the plastic bi-stable synapses switch state, as prescribed by the stochastic spike-based learning rule described in Section 9.2.7. Figures 9.10c, 9.10d show the activity of the on-chip network divided in groups of neurons during a teaching trial. The top plot shows the activity of the 128 neurons that are part of the first recurrent layer. The two last columns of the plot evidence the two pools of perceptrons that are respectively driven by a teacher-hi signal and by a teacher-low signal, depending on the input stimulus. This procedure is repeated over the entire set of training stimuli, for four different recording trials. At the end of the training procedure the state of the learning synapses is frozen, such that the pre and/or post-synaptic spikes will not cause any more transitions in the synaptic weights.

Testing the neuromorphic classifier

The testing phase is performed by presenting recordings from the same channels but for two different recording trials. The discrimination threshold, used to separate positive versus negative classification outputs is determined by re-playing all the teaching set to the classifier and by maximizing classifier's performance. In the current example, the discrimination threshold was found to be at 5.1 Hz. During test trials, the spiking activity of the two pools of neurons is averaged and if it falls above/below the threshold value it is classified as a positive/negative sample. Figures 9.11a, 9.11b show examples of successful testing of the two distinct input patterns BOS and REV. The overall performance of all testing trials is shown in Fig. 9.12. The final accuracy of this experiment is of approximately 96% on the test set: all the BOS stimuli were classified correctly and three REV stimuli were misclassified as false positives.

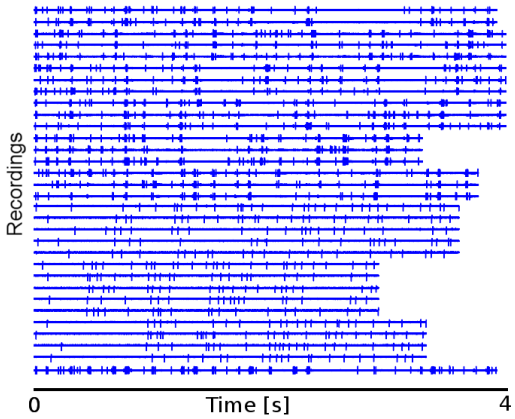
9.4 Discussions

Spiking neural networks have been shown to be able to carry out complex spatio-temporal processing and classification tasks [Giulioni et al., 2009; Maass et al., 2002; Sheik et al., 2013]. The specific topology of the neural network and the learning mechanisms adopted determine the signal processing ability of the system. These networks represent a

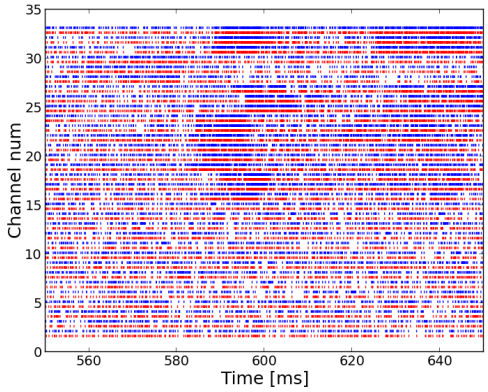
promising solution for intelligent BMIs, as they can endow them with the ability to learn to decode neuronal signals into appropriate motor commands. In particular the reservoir computing paradigm is appealing for these types of BMIs because the recurrent population of spiking neurons acts as a preprocessing stage for the readout units, which increases the range of possible functions of the input streams that can be learned by linear combinations of the read-out units. The non-linearities present in the reservoir and the inhomogeneous properties of its computing elements are beneficial for this feature [Maass et al., 2002]. In BMIs the need of extracting a diverse pool of functions is directly related to a specific computational goal, such as predicting the movement of objects or non-linearly controlling a motor actuator. From the circuit implementation standpoint, the design choice of using asynchronous circuits is advantageous because the events generated from the recorded neural signals tend to be sparse. Therefore the encoding of the addresses in AE is a good solution in term of power consumption. The mismatch and the limited precision of the analog circuits results in a diversity of responses that might be beneficial in a population coding scheme within the reservoir computing framework. This is also the case for the multi-perceptron network implemented in Section 9.3.4, in which only few perceptrons responded to the trained stimulus. As in this population coding scheme every neuron was tuned to slightly different features, we adopted the "bagging" strategy to muse many weak classifiers in parallel to improve the overall classification performance [Breiman, 1996; Skurichina and Duin, 2002]. It has indeed been argued that neuromorphic electronic circuits offer a compact and ultra low-power substrate for implementing optimal on-line learning systems [Chicca et al., 2014]. Inputs to these neuromorphic systems are typically provided by sending AER sequences of spikes from vision/auditory sensors, or produced on conventional computers. The circuits proposed in this paper can be used to create a new source of AEs that translate the activity of real neurons into the the relevant representation for further processing by neuromorphic computing cores. As recent developments are showing how to design neural networks that can optimally learn both feature extraction and pattern classification stages [Hinton, 2007], in principle it is not necessary to detect individual action potentials in the neural recording data, or to carry out elaborate spike-sorting. The circuits presented here can lead to a new generation of compact, low-power, and adaptive BMIs that can be chronically implanted to that can carry out context-dependent learning for optimally driving and controlling prosthetic devices in real-world conditions. If one has to target embedded applications, there is the requirement to develop an application-specific, low-power, and compact router. In our implementation we used an external router implemented in an FPGA board. This allowed us to easily change software and debug the AER transactions.

9.5 Conclusions

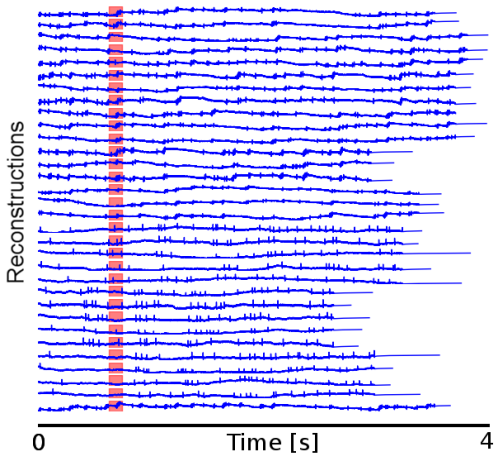
We designed, fabricated, and tested a neural recording and processing system with AER interfacing circuits suitable for transmitting bio-potentials and LFP signals to neuromorphic computing architectures. Salient features of the system are reported in Table 9.2. We interfaced this system with a reconfigurable spiking neural network architecture, endowed with learning abilities. We exploited the parallel, distributed and low-power properties of the neuromorphic architecture to design a hardware reservoir computing framework, implemented as a recurrent neural network with fixed synaptic weights. Using this network we were able to optimally decode dynamic input signals and to configure it as a feature extraction layer capable of providing input to a second layer of on-chip perceptrons. The hardware perceptrons were trained to detect a specific sequence of activations of the recurrent units in a way to respond to a specific sequence of action potentials in the input signal, while ignoring other sequences. This work offers interesting prospectives for future intelligent neural-inspired BMI.



(a)



(b)



(c)

Figure 9.9: a) Neural recordings aligned with stimulus onset. Auditory stimulus is (BOS). b) Recorded delta modulator output. Red is DN channel and blue is UP channel. c) Reconstructed recordings from Address-Event. Highlighted area shows the recording time of Fig. 9.9b, i.e.,≈ 40ms.

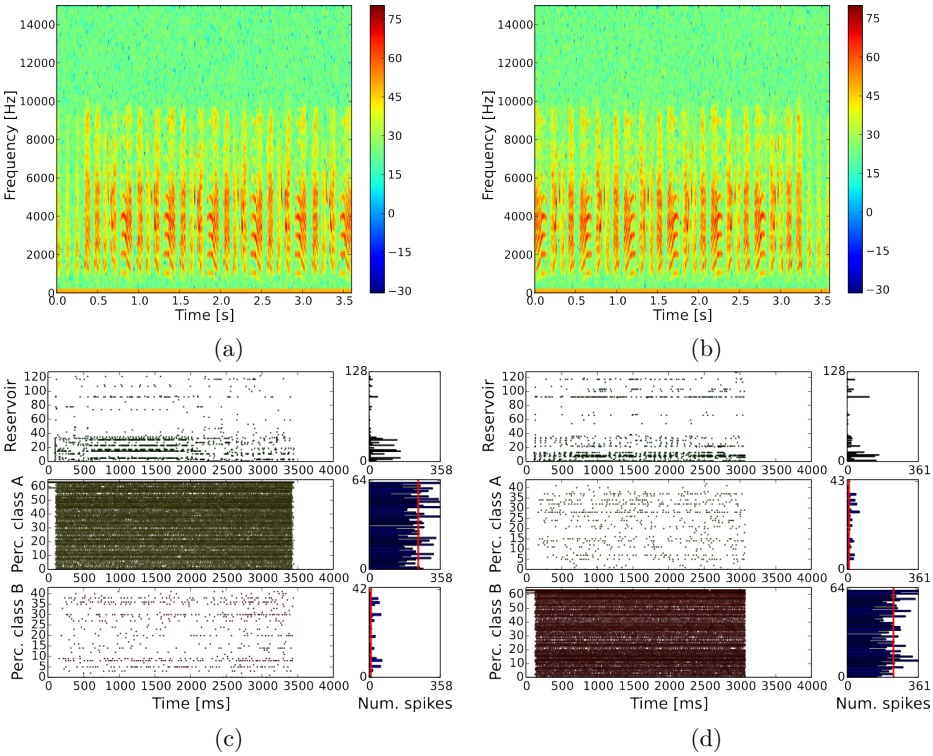


Figure 9.10: a) Auditory stimulus is bird's own song (BOS), class A. b) Auditory stimulus is reversed bird's own song (REV), class B. c) Teaching trial BOS. d) Teaching trial REV.

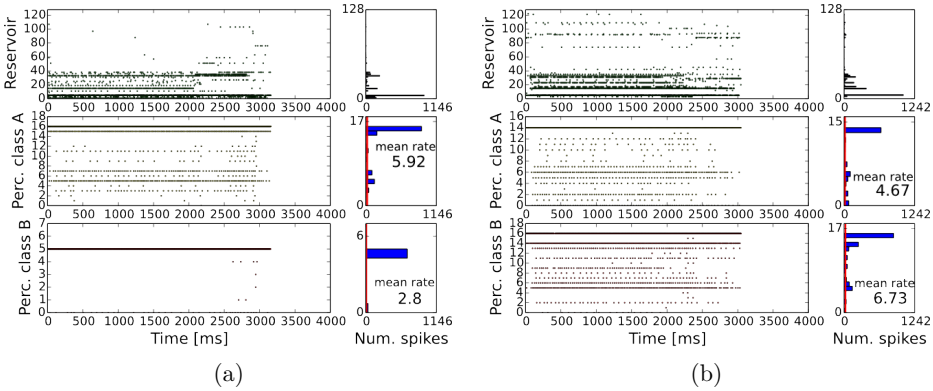


Figure 9.11: 9.11a) Typical on-chip network activity during test trial: stimulus is BOS. Every dot in the plot represent a spike; Top panel shows reservoir activity, mid panel shows activity of class A perceptrons and bottom panel shows class B perceptron activity. The plots at the right column are histograms of spike counts during the 4 seconds stimulation. 9.11b) Test trial, stimulus is REV.

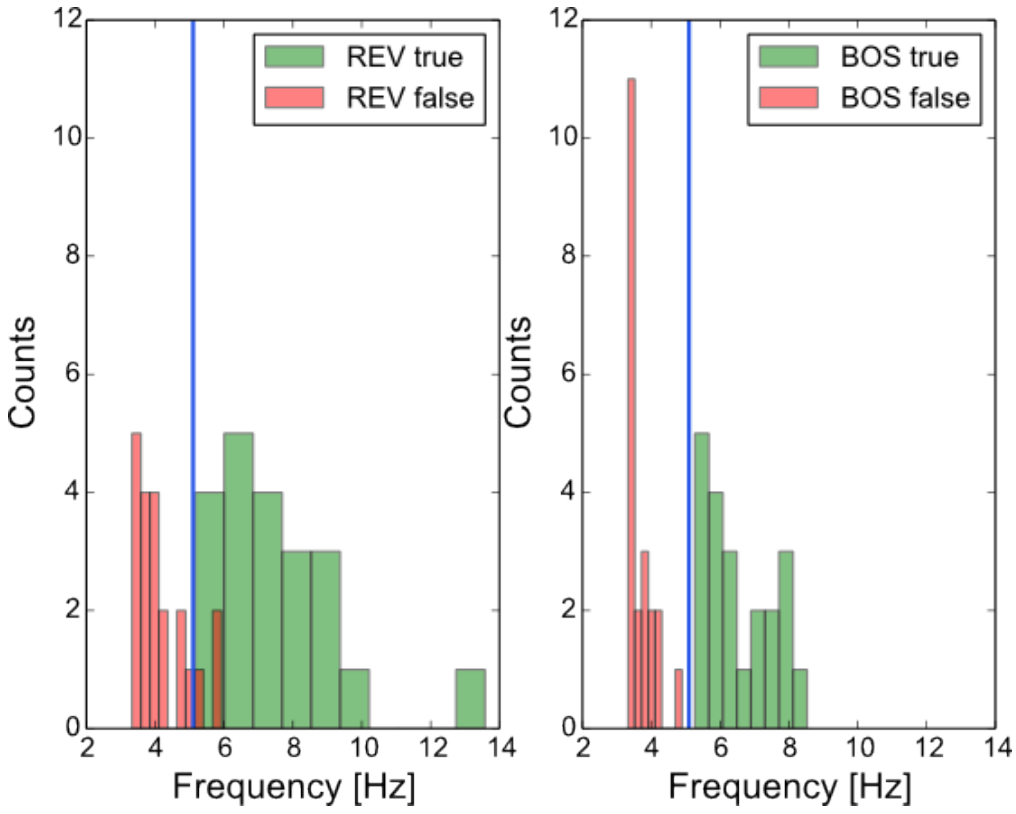


Figure 9.12: Mean rate histograms of perceptrons activity during testing. The vertical line is the discrimination threshold. Three REV trials are wrongly classified as false positive.

CHAPTER 10

Towards a neuromorphic vestibular system

F. Corradi, D. Zambrano, M. Raglianti, M. Passetti, C. Laschi, and G. Indiveri

Biomedical Circuits and Systems, IEEE Transactions on, 8:(5) 669-680, 2014

10.1 Introduction

The vestibular system plays a crucial role in the sense of balance and spatial orientation in mammals. It is a sensory system that detects both rotational and translational motion of the head, via its semicircular canals and otoliths respectively. In this work, we propose a real-time hardware model of an artificial vestibular system, implemented using a custom neuromorphic VLSI multi-neuron chip interfaced to a commercial IMU. The artificial vestibular system is realized with spiking neurons that reproduce the responses of biological hair cells present in the real semicircular canals and otholitic organs. We demonstrate the real-time performance of the hybrid analog-digital system and characterize its response properties, presenting measurements of a successful encoding of angular velocities as well as linear accelerations. As an application, we realized a novel implementation of a recurrent integrator network capable of keeping track of the current angular position. The experimental results provided validate the hardware implementation via comparisons with a detailed computational neuroscience model. In addition to being an ideal tool for developing bio-inspired robotic technologies, this work provides a basis for developing a complete low-power neuromorphic vestibular system which integrates the hardware model of the neural signal processing pathway described with custom bio-mimetic gyroscopic sensors, exploiting neuromorphic principles in both mechanical and electronic aspects.

10.1.1 Motivations

The vestibular system provides the main sensory input to our sixth sense: the ability to detect motion of the head relative to space and gravity [Cullen and Sadeghi, 2008]. It comprises two otholith organs, which detect linear accelerations, and the three semicircular canals, which detect angular velocities, over the frequency range of the natural movement (i.e., approximately 0–20 Hz). These sensory structures are positioned in the inner ear and provide us with a reference frame for the movements and orientations in space [Sadeghi et al., 2007], [Goldberg et al., 1984]. This information is involved in several neural pathways for sensory integration or control systems, to maintain both head and body posture, such as the balance control system, and to compensate for head movements around the visual axis, for stabilizing vision while adapting to a wide range of visual dynamics [Itō, 1984]. In general, the vestibular system plays a key role in the definition of our sense of movement [Berthoz, 2002]. This work aims at developing a physical model of the vestibular system by emulating the response properties of the vestibular afferents via biophysically realistic electronic circuits interfaced to a commercial IMU. In particular, we present an implementation of a neuromorphic vestibular system built by connecting the IMU to a custom VLSI neuromorphic device that comprises silicon neurons and synapses able to reproduce biologically realistic dynamics [Chicca et al., 2014]. This system mimics the spiking response of a vestibular system, providing the relevant information content useful for later processing stages (e.g., to stabilize vision, maintain posture, etc.). Although several artificial sensors have been used as inertial measurement units to model the dynamics of the vestibular system, only few attempts have been proposed that directly emulate the detailed properties of the real biological sensors. Recently custom devices have been proposed as efficient implementations of gyroscopic sensors [Andreou et al., 2013], which could be used in conjunction with the neuromorphic electronic models presented in this paper. Here we focus on modeling the post-processing neural circuits of the vestibular system, independently from the technology used to measure the inertia. In particular, we use neuromorphic electronic circuits to reproduce the information provided by the biological vestibular system and validate the approach by comparing our real-time circuit measurements to data derived from computational neuroscience models. The method we propose aims both at understanding the properties of real vestibular systems and at implementing it in compact electronic hardware for embedded systems that can be used in neuroscience research investigations, in practical robotic applications [Dario et al., 2005], and eventually in the neuroprosthetics domain, for providing sensory inputs that can be used to help restore the ability to maintain balance [Constandinou et al., 2008a,b] using signals and representations that are as close as possible to the real ones. In the robotic and conventional electronics domain, the approach of using IMUs to

stabilize moving platforms and systems has already been adopted in the past, even using bio-inspired methods [Dean et al., 2002; Delbruck et al., 2014; Franchi et al., 2010; Shibata and Schaal, 2001]. However these approaches have been mainly restricted to the domain of gaze stabilization and Vestibulo-Ocular Reflex (VOR) modeling, and have not considered the computational properties of a complete embodied physical implementation of the vestibular system. In neuromorphic engineering, neurons and synapses are implemented as electronic circuits that use the physics of silicon to directly emulate the electrophysiological behaviour of their biological counterparts [Indiveri et al., 2011]. In general, the neuromorphic engineering approach radically changes the signals provided by the sensory devices realized, leading to new brain-inspired and alternative methods of signal processing. For example, spike-based neuromorphic vision and auditory sensors (e.g., silicon retinas, and silicon cochleas) [Delbruck et al., 2010b; Liu and Delbruck, 2010] provide in output a continuous asynchronous stream of events that represent the spikes generated in real-time by the sensor pixels or auditory channels, as these pixels sense relevant information. These sensors therefore are characterized with very high temporal resolution, and typically with very sparse activity (i.e., with very low bandwidth requirements). This is a major difference from conventional sensors that operate on frames, and take "snapshots" of the visual or auditory scene at regular intervals (e.g., every 30 ms), irrespective of the signals being sensed. The neuromorphic vestibular system we propose is not characterized by similar low-power performances as the mentioned neuromorphic systems, however, in principle, it could provide an efficient spike-based coding of the vestibular information for such robotic applications based on highly parallel event-based computational architectures. To demonstrate the usefulness of this approach in practical applications, we applied the system to implement a modular head-direction integrator network which integrates information about angular velocities and keeps track of the current angular position in a stable self-sustained memory state [Amit, 1992]. Much work has been done in modeling the head-direction movement system that is responsible for maintaining an accurate representation of the heading information [Hahnloser, 2003; Xie et al., 2002; Zhang, 1996]. We ground our work on these models, and present a first implementation of a full system in which the spiking input is generated by a sensor that mimics the information transmission of the biological vestibular system. In the next sections we describe the main elements of the sensor developed, starting with a description of the model used. We then describe the experimental set-up used and present the experimental results. Although the neuromorphic vestibular system we present is only an approximation of the real biological system considered (because of the smaller number of neurons used in the artificial system), the experimental data we provide demonstrates the validity of the followed approach, as a faithful model of the biological counterpart. An

early implementation of this work was presented in [Passetti et al., 2013]. However, that system modelled only the semicircular canals, while in this work we implement physical models of both the semicircular canals and the otholitic organs, provide results based on angular velocities around three orthogonal rotational axes and on linear accelerations in two orthogonal planes, and present the head-direction detector network implemented as an integrator neural network.

10.2 A Neuroscientific Model of the Vestibular System

We present a system that implements a neuromorphic model of the otolith organs, the semicircular canals and an integrator network (see Fig. 10.1 and Fig. 10.2). The system aims at emulating the spike timing response of the sensory neurons (the hair cells) present in the vestibular afferents [Cullen and Sadeghi, 2008] using a neuromorphic VLSI device that comprises silicon neuron and synapse circuits able to reproduce biologically realistic dynamics with biologically plausible time-constants [Chicca et al., 2014]. To demonstrate a practical application of the system proposed, a head-position network is implemented in hardware using spiking neural network to resolve ambiguities and maintain state (see Fig. 10.2). As the information carried out by the artificial vestibular system is fully characterized, we do not model all the thousands of afferents present in the biological system, but emulate only a small subset of them. Afferents in this system are characterized as either *regular* or *irregular* on the basis of their resting discharge, which is correlated to distinct morphological features at their peripheral terminations [Goldberg et al., 1984; Sadeghi et al., 2007]. Sadeghi et al. (2007) showed the role of variability into information transmission, suggesting that the two coexistence of the neuronal populations in the peripheral sensory system has played an important functional role. The hypothesis is that regular afferents rely on their precise spike timing for transmitting information (therefore using a temporal code), while the irregular afferents use their mean firing rate (therefore using a rate code). As suggested in [Sadeghi et al., 2006], irregular afferents best encode information for high frequencies whereas regular afferents transmit information about the detailed time course of the stimulus. As the vestibular otolith model is responsible of encoding linear accelerations, the model neurons are distributed along two orthogonal planes: a vertical plane Π_v, and a horizontal plane Π_h. Neurons in these planes are arranged in a grid in which the position of the cells indicates their preferred stimulus orientation. For example, neurons placed along the positive abscissa are tuned for positive accelerations in the x directions. The system modeling the semicircular canals encodes information about angular velocities. Therefore we realized a distributed network in three dimensions. We implemented a pool of four neurons for each plane, as shown in Fig. 10.1. Among these four neurons, two are responsive to angular velocities for clockwise rotations

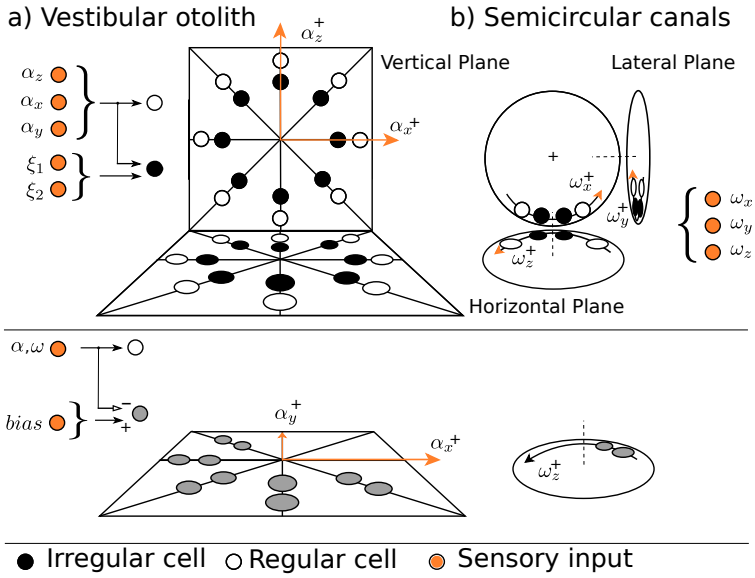


Figure 10.1: **Schematic view of the implemented model** a) Vestibular otolith organs provide information about linear accelerations in two orthogonal planes and in specific preferred directions. b) Semicircular canals provide information about angular velocities in the three orthogonal rotational axes. Bottom) A bias has been added to the sensory information for the highlighted cells.

and other two are responsive to angular velocities for counterclockwise rotations. The hair cells in the vestibular system are oriented in several preferred directions, thus a stimulus is encoded by different cells excited or inhibited according to their orientations. In our model, we used the same sensory signal as input for the vestibular neurons placed as shown in Fig. 10.1. The negative component of the vector (represented as gray neurons in Fig. 10.1, bottom) has been modeled as a inhibitory input for a bias neuron. The integrator network that implements the head-direction detector model is shown in Fig. 10.2. This network integrates information coming from the semicircular canals and stores the angular position in a memory layer of cells. Every circle in Fig. 10.2 represents a neuron; only connections for filled coloured neurons are shown entirely and the central column of neurons is repeated in the network and distributed in a closed ring. The top layer shows neurons in the semicircular canal, these cells receive information encoding the angular velocity ω_x. Four cells are implemented: two per direction, one regular and one irregular. These semicircular canal direction selection cells make excitatory projections to two distinct groups of neurons: increment cells and decrement cells (move memory). All neurons that are part of the increment and decrement pools are inhibited by the active memory position neuron ϕ_x, except the ones directly below the active memory neuron,

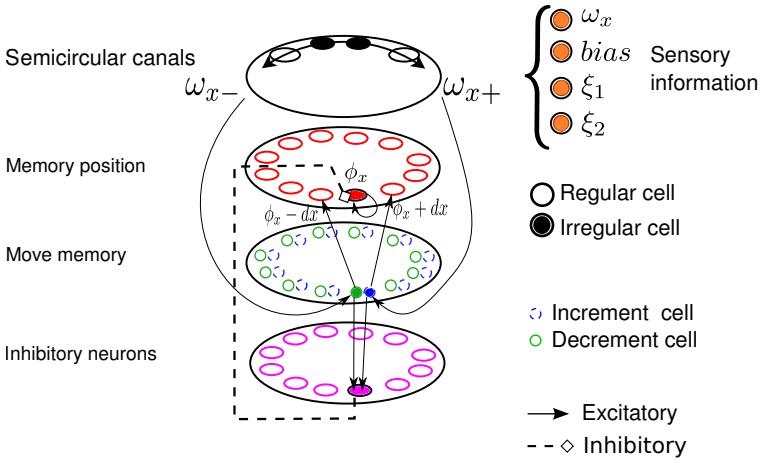


Figure 10.2: **Integrator network.** The top layer shows a semicircular canal pool of neurons that encodes information about rotations around the x axis. The activity of the semicircular canal cells is projected to the move memory layer. The move memory cells are inhibited by the activity of the current memory position cell, except the filled ones (green and blue, inhibitory connections from the memory position layer to the move memory cells are not shown in the schematic). The memory position cells are capable of sustaining persistent activity via recurrent connections. In the move memory layer increment and decrement cells make precise projections to the memory position cells, i.e. the increment (decrement) cells project to the right (left) of the current position. When move memory cells receive a strong input they activate the adjacent memory position as well as the inhibitory neuron that inhibits the previous memory position. This network integrates information coming from the semicircular canal and store the current angular position in the memory layer (red).

gray filled color in the schematic (inhibitory connections are not shown for clarity). When sensory information is present at the input of the network, i.e., the sensor has been rotated along the x axis, the respective increment or decrement neurons are excited. These increment and decrement neurons make excitatory projections to the adjacent top neuron $\phi_x + dx$ or $\phi_x - dx$, which will be able to fire persistently via its recurrent connections. The activated increment or decrement neuron will also excite the bottom inhibitory neuron which in turn will inhibit the previous active memory position neuron. This network integrates the angular velocity input and stores its current value in memory. The memory is represented by the self-sustained activity of a neuron in the memory layer. The neuron model of the vestibular afferents is a modified leaky integrate-and-fire neuron with a dynamic threshold, as described in [Sadeghi et al., 2007]. The neuromorphic silicon neuron circuits used to emulate this model in real-time are the ones described and fully characterized in [Chicca et al., 2014]. The theoretical model equations are reported as Eq. 10.1 and 10.2. These equations are used to map the model parameters to the current and voltage biases that determine the response properties of the silicon neurons and synapses. The hair cells of the receptor organs mainly respond to a preferred orientation. At resting state, these cells are active with a resting discharge with low rate of variability for the regular cells and high rate of variability for the irregular ones. A deflection in the preferred direction causes depolarization of the membrane potential (the firing rate increases) and a deflection in the opposite direction causes its hyperpolarization (the firing rate decreases). The afferent cells coming from the three canals encode head velocities in the three mutually orthogonal axes. In order to complete the information representation, two opposite cells (one regular and one irregular) for each axis are added as showed in Fig. 10.1. In case of otolith organs the biological sensor redundancy guarantees that there is at least one afferent for any possible direction [Angelaki and Dickman, 2000]. In the presented system 8 directions for each plane are implemented (see Fig. 10.1). We did not take into account the individual biomechanics and sensitivity to inertia of biological hair cells but, due to the mismatch in integrated circuits fabrication, significant variability can be obtained by connecting the same transducer to different silicon afferent neurons. The membrane voltage v and threshold w, over times between action potentials, obey the following differential equations (note that the model emulates only the spike timing response and that voltages and currents are dimensionless):

$$\dot{v} = -\frac{v}{\tau_v} + \frac{I_{\text{synap}}}{\tau_v} \tag{10.1}$$

$$\dot{w} = (w_0 - w)/\tau_w \tag{10.2}$$

Table 10.1: Model parameters for regular and irregular afferents: (a) shared parameters; (b) individual parameters.

	Common
τ_v (ms)	1
τ_w (ms)	9.5
w_0	0.05
T_{refrac} (ms)	1
τ_a (ms)	20

	Regular	**Irregular**
I_{bias}	0.0515	0.049
Δw	0.003	0.001
σ	0.00007	0.0015
G_H (ms/deg)	0.0156	0.0315
G_A (ms/deg)	0	0.0315

where τ_v is the membrane time constant, τ_w is the threshold recovery time constant, and w_0 is the equilibrium value for the threshold. When $v(t) = w(t)$ an action potential is produced, and w is incremented by Δw, while v is reset and forced to 0 for T_{refrac} ms. The resulting threshold is therefore self adapting to repetitive firing by hyperpolarization, with an upper bound dependent on T_{refrac}. In [Sadeghi et al., 2007] it has been demonstrated that the internal noise is a major factor contributing to the differences between the resting discharge of the two cells. Thus, it can be modeled as a gaussian input added to inertial measure. The synaptic current I_{synap} is:

$$I_{\text{synap}} = G_H HV(t) - G_A X_A(t) + I_{\text{bias}} + \sigma\xi(t) \tag{10.3}$$

$$\dot{X}_A = -\frac{X_A}{\tau_A} + \frac{HV(t)}{\tau_A} \tag{10.4}$$

where $HV(t)$ is the head rotational velocity with respect to the considered axis, I_{bias} is a constant bias current, $\sigma\xi$ is Gaussian white noise with zero mean and SD σ and G_H is the intensity of the signal. In case of irregular afferents the function $HV(t)$ is low-pass filtered (with cut-off frequency 50 Hz $= 1/\tau_a$) to obtain X_A, which is then subtracted with a gain G_A from $G_H HV(t)$. Table 10.1 shows the model parameters. The parameters have been selected to be within the biological range [Smith and Goldberg, 1986; Yakushin et al., 2006], and in are in accordance with the parameters used in [Sadeghi et al., 2007].

Refractory period T_{refrac} and *Spike frequency adaptation* are important features of the model; neuron's adaptation rate has time constant τ_w (see Table 10.1). In Fig. 10.2a the

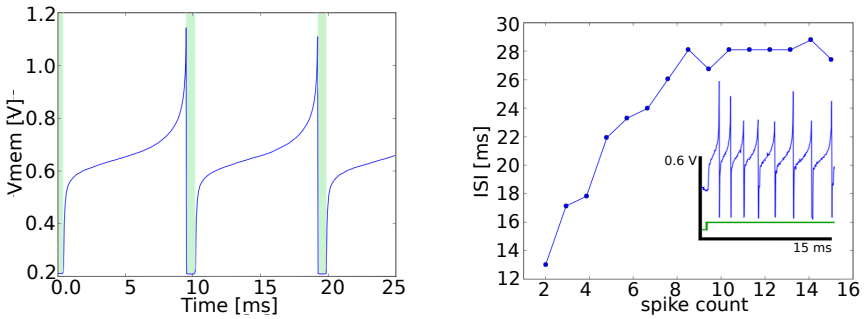


Figure 10.3: **Neuron calibration** (a) neuron membrane potential. The shaded area shows the refractory period $T_{\text{refrac}} = 1$ ms. (b) Spike frequency adaptation. The neuron settles into an adapted state after about 7 spikes. The figure inset shows the membrane potential over time. The bottom trace in the inset represents the input current step. Figure from [Passetti et al., 2013]

.

shadowed areas highlight the refractory period, while in Fig. 10.2b we show the frequency adaptation effect, in response to a current step input, occurring after about 10 ms as in [Passetti et al., 2013].

Information transmission analysis

In order to characterize quantitatively the response properties of our system and compare them to biological data, we performed information transmission analyses. The results of these analyses are presented in Section 9.3. Responses to random stimuli were characterized using three distinct measures: (1) the normalized Coefficient of Variation (CV) of the neuron output ISI, (2) the Coding Factor (CF), and (3) the mutual information rate density (MI) between reconstructed signal from spike train and stimulus (head velocity). The CV* is computed as the normalized ratio between the standard deviation and the mean of the ISI distribution [Goldberg et al., 1984]. To estimate the coding factor as in [Sadeghi et al., 2007] we reconstructed the sensor's raw output measurement from the spike train $r(t)$, using the stimulus-reconstruction technique described in [Rieke, 1997]. Once the reconstruction is made, we estimated the Root Mean Squared-Error (RMSE) between reconstructed input stimulus and actual stimulus. The coding factor is then calculated as: $CF = 1 - \epsilon/\sigma_{stim}$ where ϵ is the RMSE and σ_{stim} is the standard deviation of the stimulus. To evaluate the mutual information rate density, a first step is the calculation of the coherence $C(f)$ between the head velocity and the spike train $r(t)$,

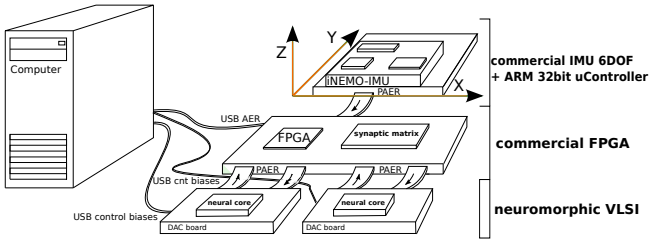


Figure 10.4: **Schematic view of the setup, with inertial sensor, communication logic, and mixed signal neuromorphic devices.** A commercial IMU is placed on a mechanical platform which provides controlled linear accelerations and rotations. The IMU produces spikes from nine channels which encode angular velocities and linear accelerations. The spikes coming from the IMU channels are routed to two neurmorphic VLSI chips by a commercial FPGA board. The spikes produced by the system are monitored by a computer via a standard USB connection. Additional USB connections are used to set the analog parameters that configure neurons' and synapses' properties on the neuromorphic chips.

calculated as $C(f) = |P_{rs}|^2 / (P_{ss}(f)P_{rr}(f))$ where P_{rs} is the cross spectrum between stimulus and response, P_{ss} is the power spectrum of the stimulus, and P_{rr} is the power spectrum of the spike train. The computation of the spike train power spectrum is achieved converting the spike train into a binary sequence where value 1 is assigned to each action potential and the spike train power spectrum is computed using multitaper estimation techniques [Jarvis and Mitra, 2001]. A bound estimate of the mutual information rate density is derived from the coherence $I(f) = -log_2 [1 - C(f)]$ [Rieke, 1997], normalized by the mean-spike rate $MI(f) = I(f)/f_r$ [Borst and Haag, 2001]. The mutual information rate density $MI(f)$ results in bits/spikes/Hz, we integrate the mutual information rate density over the frequency range (0-20 Hz).

10.3 Material and methods

In this section we describe the experimental setup used to implement the vestibular system and to measure all state variables and control signals for characterizing it. We first describe the custom configuration of the IMU used to represent the sensory information of Six Degrees of Freedom (DOF) as "spikes" (i.e., digital output pulses). Then we describe how these spikes are encoded using the AER protocol and how they are mapped to the neuromorphic chips.

The sensor

The flow of the signal starts with the sensory information which is collected by the IMU. The IMU inertial transducer produces digital outputs representing angular velocities and linear accelerations, relative to three orthogonal coordinate axes. Information is encoded using a pulse-frequency modulation scheme, and transmitted using the AER protocol. In addition to a Micro Electro Mechanical System (MEMS) inertial transducer, the IMU comprises an Advanced RISC Machine (ARM) core micro-controller. The overall system (commercial iNEMO M1 system on board, produced by STMicroelectronics[1]) is used to produce nine distinct outputs: three for the linear accelerations, three for the angular velocities, two for noise components (used to simulate more realistic bio-sensors) and one for an external bias signal. All nine outputs represent the intensity of the signal they encode as Address-Events, using a pulse frequency modulation scheme: the pulse frequency of each inertial transducer sensory channel (f_c) encodes linearly the respective sensor's output in a range of frequencies between $f_{\min} = 400\,\text{Hz}$ and $f_{\max} = 800\,\text{Hz}$ (corresponding respectively to the highest negative and positive measured velocity/acceleration):

$$f_c = \frac{f_{\max} - f_{\min}}{2|Limit_c|} Reading_c + \frac{f_{\max} + f_{\min}}{2} \tag{10.5}$$

where c is one of the α_x, α_y, α_z, ω_x, ω_y or ω_z channels, corresponding to linear accelerations and angular velocities on the three axes respectively. The *Limit* variable represents the limit value for the input range at which saturation of the output frequency occurs. The *Reading* variable is the actual raw value returned by the sensor. The sensor's full scale of $\pm 250\,\text{deg/s}$ has been used as the *Limit* angular velocity for the ω channels. Different full scale values have been used, instead of the ones provided by the accelerometers, on the iNEMO for the α channels. A maximum of $\pm 0.6\,\text{g}$ has been set for the frequency linear mapping. These values are compatible with the linear range of the biological system response [Angelaki and Dickman, 2000]. The inertial sensor sampling frequency has been set to $100\,\text{Hz}$, while the full-scale input range is $\pm 2\,\text{g}$ and $\pm 250\,\text{deg/s}$ for the accelerometer and the gyroscope respectively. The gyroscope's nominal sensitivity is $8.75 \cdot 10^{-3}\,\text{deg/s/digit}$, and its output is digitally low-pass filtered with $25\,\text{Hz}$ cut-off frequency; the accelerometer has nominal sensitivity $10^{-3}\,\text{g/digit}$ and is not filtered. Digital communications with the gyroscope are managed by the Serial Peripheral Interface (SPI), while communication with the accelerometer is handled by a High Speed Inter Integrated Circuit protocol (I^2C). All communications are performed in interruptible mode. Higher priority has been given to the generation of a steady output on the

[1] URL: `http://www.st.com`

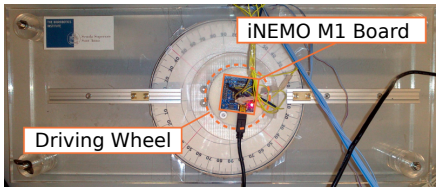


Figure 10.5: **Mechanical setup for angular velocities.** The motor, mounted below the driving wheel, is coaxial with the gyroscope on the iNEMO M1 Board.

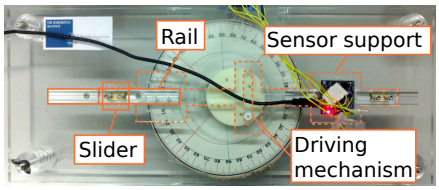


Figure 10.6: **Mechanical setup for linear accelerations.** The driving mechanism transforms the uniform circular motion of the motor (mounted in the same configuration as in Fig. 10.5) in a one axis sinusoidal linear motion for the sensor support and the sensor itself. The constrained axis is determined by the two sliders and the rails. Different orientations for the sensor are provided on the support.

timings signals, interleaving low priority communications with the sensors to acquire new readings. The two channels used for the noise components are controlled by the ARM micro-controller hosted on the IMU board: the noise sources have been simulated by generating distinct Gaussian spike distributions on two different source channels (see ξ_1, ξ_2 in Fig. 10.1). A Gaussian distributed array of $32k$ samples, corresponding to a spike frequency with $\mu = 500$ Hz and $\sigma = 100$ Hz, has been pre-generated, stored in the MCU's static memory and used as a Lookup Table. The delay between two consecutive pulses on the same noise neuron channel is chosen by taking a pseudo-random value from the array with a uniform distribution. The external bias signal produces a constant-frequency spike train that is generated at a maximum output frequency of 800 Hz. This signal is used as an additional inhibitory input to neurons having at least an inhibitory synapse (grey neurons in Fig. 10.1 bottom). This off-chip bias allows lower spike rates as the inhibitory input rises. The connection between the sensory system and the neuromorphic chip has been achieved via the FPGA, and implemented using a parallel AER protocol.

The mechanical setup

In order to obtain results comparable with the literature, a controllable angular velocity and linear acceleration input signal for the sensor was needed. Regarding the semicircular canals, the sensor was mounted directly on the driving wheel, coaxial with the motor

axis (Fig. 10.5). This resulted in a constant angular velocity around the vertical axis. A sinusoidal acceleration on one axis of the horizontal plane, for the otoliths, has been achieved by the mechanical setup shown in Fig. 10.6. The continuous current reduced Faulhaber motor[2] produces uniform circular motion at a frequency depending on the supplied voltage. The circular motion is transformed in a linear acceleration by the driving mechanism, the mechanical platform is capable of producing sinusoidal accelerations between $\pm 5.7\ \ m/s^2\ \approx \pm 0.57G$ ($G = 9.81\ \ m/s^2$).

10.4 The neuromorphic VLSI multi neuron chip

The neuromorphic component of the system comprises two instances of a mixed signal VLSI chip that integrates silicon neuron and silicon synapse circuits that directly emulate the properties of their biological counterparts [Chicca et al., 2014]. Depending on how input and output spikes are routed to the synapses, via a "synaptic matrix" look-up table managed by an external FPGA board (see Fig. 10.4), the chips can implement arbitrary spiking neural network topologies. The spikes are encoded as digital pulses, transmitted asynchronously via parallel AER communication. The neuromorphic chip was fabricated using standard $0.18\,\mu m$ CMOS 1-poly 4-metal process. Each chip contains 58 adaptive exponential Integrate-and-Fire neurons with biophysical realistic dynamics, implemented in analog sub-threshold circuits, that have been fully characterized in [Chicca et al., 2014; Indiveri et al., 2011]. Each neuron has a dendritic input which comprises 32 programmable synapses with four bit resolution weights stored in a local asynchronous Static Random Access Memory block, as well as 8 synapses with spike-based plasticity circuits. The programmable synapse circuits and asynchronous SRAM block have been already presented in [Corradi et al., 2014b; Moradi and Indiveri, 2011], while the learning synapses have been described in [Mostafa et al., 2014]. In here we focus on the system-level aspects and point the interested reader to the above mentioned publications for details on the individual circuits. The configuration of the analog circuit properties is achieved by setting the circuit voltage biases via a DAC board on top of which the neural chips are hosted. All the analog parameters are shared among the same neural core chip, but the two distinct cores can have dissimilar parameter configurations. Depending on the biases chosen, the neuron circuit is capable of reproducing a wide range of spiking behaviours and the dynamics of the membrane potential is well approximated by the adaptive exponential Integrate-and-Fire model [Brette and Gerstner, 2005]. The circuit biases were chosen to match those of the model described in Section 10.2. We configured the system (by setting the constant I_{bias} term) such that when the sensor is in a firm

[2] url: http://www.faulhaber.com/uploadpk/EN 2342 CR DFF.pdf

position all channels encoding linear accelerations as well as angular velocities emit spikes at a constant rate of 600 Hz (the middle value of the linear mapping). The output of these channels is then directed to 44 silicon neurons in one of the two neural core chips. This pool of 44 neurons is composed of 32 neurons for the vestibular otolith organs and 12 neurons for the semicircular canals afferents (see in Fig. 10.1). The sensor's outputs encoding linear accelerations in the space (x, y, z) have been mapped to 32 vestibular otholithic neurons lying on two orthogonal planes, the horizontal plane Π_h, and the vertical plane Π_v, according to:

$$w^i_{\Pi_h} = \alpha_y \cdot sin(\phi) \cdot w_{max} + \alpha_x cos(\phi) \cdot w_{max} \qquad 1 < i < 16 \quad (10.6)$$

$$w^i_{\Pi_v} = \alpha_z sin(\theta) \cdot w_{max} + \alpha_x \cdot cos(\theta) \cdot w_{max} \qquad 16 < i < 32 \quad (10.7)$$

where w^i is the 4 bit weight of the synaptic contact between the sensors' output channels $\alpha_x, \alpha_y, \alpha_z$ and the i-th neuron. The first line in eq. 10.6 refers to the sensor's output mapped on the horizontal plane while the second term maps the sensor's output on the vertical plane (as in Fig. 10.1). ϕ is the neuron displacement angle on the (x, y) plane, while θ is the displacement angle on the (y, z) plane. Finally $w_{max} = 15$ represents the maximum digital weight value. The sensory information about the angular velocities has been mapped to 12 neurons using fixed synaptic weight connections. Neurons lying in the second, third, and fourth quadrant of the planes Π_h, Π_v and neurons encoding anti-preferred directions in the vestibular system are connected to the excitatory channel bias output: therefore these connections are mapped to an inhibitory synapse, making neurons lower their spike rates as the inhibitory input rises. With this mapping we achieved a clear distinction between preferred and anti-preferred directions.

Calibration of the neuromorphic VLSI chips

Since in one neural chip all the analog voltage parameters are shared among all neurons, and because of the mismatch caused by the process variations, the application of the synaptic weights given by eq. 10.6 did not produce accurately the expected tuning curves. We therefore applied a further calibration method to overcome this problem: the method that consists of measuring the actual response function of all neurons in the device while sweeping all digital synaptic weight values and determining a mapping between the desired weight and the actual weight value setting [Corradi et al., 2014b]. This measurement can be carried out in parallel for all neurons using synthetic spike trains produced by a standard computer. In particular, distinct regular synthetic spike

trains have been generated simulating each sensor's output at the fixed frequency of 600 Hz. Once all the response functions were measured, we used a least square estimates method to extrapolate a digital value that best matches a desired synaptic weight. The calibration method converged when the neurons fired at about 100 Hz in the absence of external stimulation, as observed in real neurons present in the vestibular-nerve afferent cells [Sadeghi et al., 2007]. The term $\sigma\xi(t)$ in eq. 10.4 is used to separate regular and irregular cells. By connecting the Gaussian noise channels outputs to irregular neurons we achieved a clear distinction among the two classes of cells. The two Gaussian noise channels outputs have been connected to two distinct fixed synapses, one excitatory and one inhibitory. Figure 10.5 shows the properties of regular and irregular cells configured in this way. In Fig. 10.5 we show the Power Spectral Density (PSD) the of neuron's spontaneous activity for regular and irregular conditions. In the regular condition the neuron shows low variability in its ISI. This is reflected in the PSD visible in the left column of Fig. 10.5, which contains peaks at the resting state (~ 100 Hz) as well as integer multiples representing higher harmonics. On the other hand, the neuron's response in the irregular condition showed a much less structured PSD with only a clear peak at the fundamental frequency (~ 100 Hz, mean rate spiking frequency). Another indicator of the neuron's firing rate properties is the CV (as defined in Section 10.5): this measure highlights the neuron's variability at resting state, and explains differences between the two spectra densities at low frequencies (see Fig. 10.5).

10.5 Results: information processing in the neuromorphic system

Resting discharges of the regular and irregular silicon neurons (shown in Fig. 10.5) are compatible with CV biological data presented in [Sadeghi et al., 2007], where a neuron is said to be *regular* for coefficients of variation $CV^* < 0.15$ and *irregular* otherwise. In our case we obtained coefficients of variation $CV^*_{reg} = 0.002 \pm 0.001$ and $CV^*_{irr} = 0.45 \pm 0.15$ for the regular and irregular neuron respectively. To evaluate the information carried by the vestibular afferent spike train, we characterized the response of the system to three random movements, by reconstructing the sensor output from the spike train using the technique presented in [Rieke, 1997]. The quality of the reconstruction is related to the information carried in the vestibular afferent spike train. Figure 10.5 shows the measured head velocity signal along with the signal reconstructed from the relevant spike train. As expected, regular afferents follow the head velocity signal with more accuracy than irregular afferents. In Table 10.2 we report relevant measurements, defined Section 9.3, that characterize the information transmission. These measurements are compared and found to be in agreement with their corresponding values from in vivo experiments with macaque monkeys [Sadeghi et al., 2007]. The tuning of silicon neurons is approximately

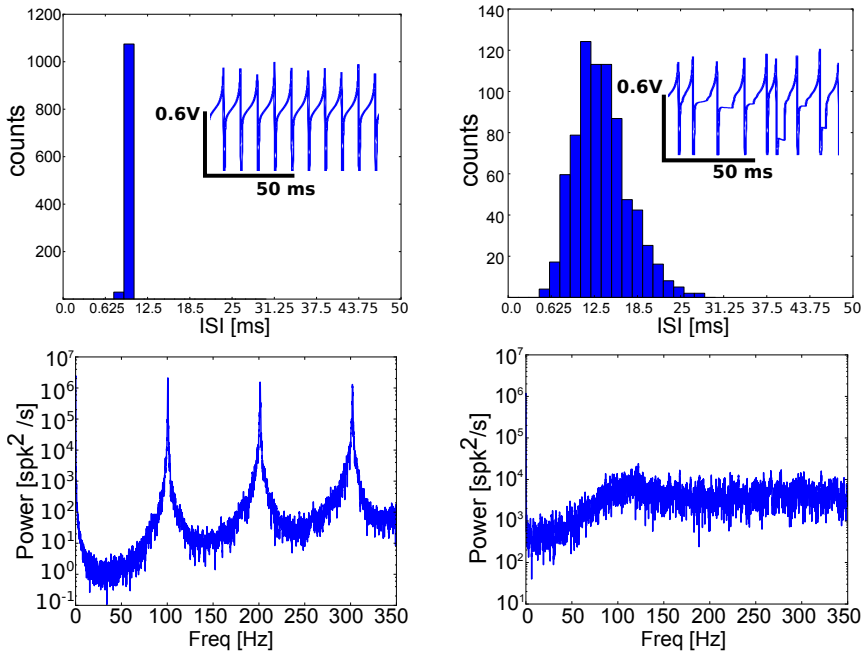


Figure 10.7: **Measured membrane potentials, Inter-Spike Interval (ISI) distributions, and power spectra density analysis** (a) Regular neuron ISI histogram at rest (spontaneous activity condition). The inset shows a snapshot of the membrane potential recorded from the chip. (b) neuron ISI for the irregular spontaneous activity condition at rest. Sub-figures: (c), (d) shows the PSD for regular and irregular neurons. Note that the regular neuron has three peaks in the PSD. By contrast, irregular neuron shows an higher energy distribution at low frequencies.

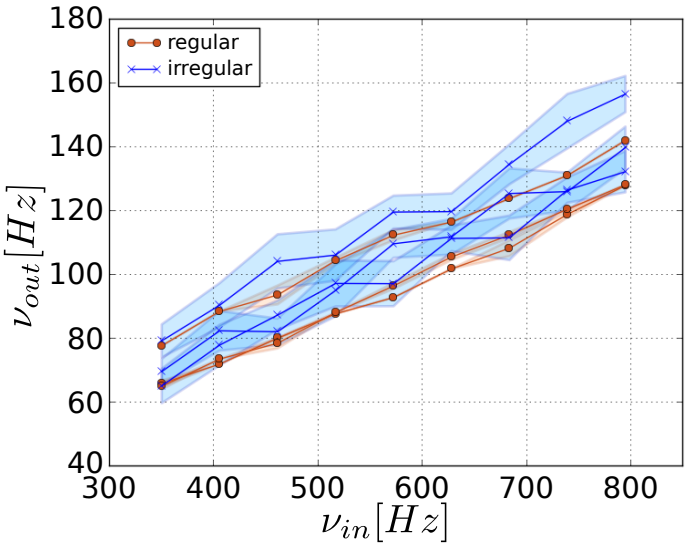


Figure 10.8: **Input/output characteristics of the vestibular afferents:** Firing rate measured from 6 neurons: regular (circles) and irregular (crosses). Shaded areas represent the standard deviation of four different stimulations. Adapted from [Passetti et al., 2013].

linear in the spiking response over the frequency range [350 - 800] Hz (see Fig. 10.8). Figure 10.9 shows how the regular and irregular neurons encode the angular velocity measurements, demonstrating that the information is well preserved for a regular neuron, and how the added noise affects the firing rate of the irregular neuron. Figure 10.9 shows how the regular and irregular neurons encode the angular velocity measurements, demonstrating that the information is well preserved for a regular neuron, and how the added noise affects the firing rate of the irregular neuron. As shown in Fig. 10.11, two neurons are tuned for two orthogonal axes, i.e., the first has a non-zero weight for the x-axis acceleration synaptic input only, the second for the y-axis acceleration. The figure

Table 10.2: Summary of results

	Neuromorphic System	**Primates ([Sadeghi et al., 2007])**
$CV_{reg}*$	0.002 ± 0.001	0.08 ± 0.03
$CV_{irr}*$	0.45 ± 0.15	0.34 ± 0.12
CF_{reg}	0.61 ± 0.06	0.39 ± 0.13
CF_{irr}	0.20 ± 0.12	0.24 ± 0.08
MI_{reg} (bits/spike)	0.46 ± 0.10	0.36 ± 0.25
MI_{irr} (bits/spike)	0.24 ± 0.07	0.18 ± 0.08

shows the response of the two neurons applying the same sinusoidal force to the platform rotated in eight different positions by 45 degrees each. The connections' weights of the neurons are calibrated in such a way that the afferent firing rate at rest corresponds to about 100 Hz output firing rate; the weights assigned to the α_x, α_y linear acceleration components on the plane for the two different neurons originate the variations for the plots in green and blue, with 90 degrees phase displacement. The continuous lines in Fig. 10.11 are obtained by fitting a sinusoid function to the Inter-Spike-Interval. The normalized root-mean-square error is shown in every plot. Such a difference between the sinusoid function and the actual behavior of neurons might be caused by the non perfect sinusoidal oscillation of the mechanical platform as well as noise and mismatch in the system. Another important aspect of these tuning curves is their amplitude as a function of stimulus orientation. Figure 10.12 shows the fitted amplitude of the two neurons as a function of stimulus orientation. The data present a clear minimum for the green plot while the blue plot has a maximum, and vice-versa. As an application, using the spike-encoded output from the gyroscope we implemented a ring neural network to integrate angular velocities and keep track of the platform's rotation angle (see Fig. 10.2). Angular position is encoded with 30 deg resolution by 12 neurons (see the red circles in Fig. 10.2). The integrator works as follows: suppose only the red-filled memory position neuron in Fig. 10.2 is firing at the beginning; it is capable of sustaining persistent activity due to a strong recurrent connection. As the platform starts rotating left (right) neurons in the move memory layer receive an excitatory input; all neurons in the move memory layer are inhibited by the active memory position neuron, except for those represented immediately below it, which start firing and excite the next position neuron, that encodes an angle increment (decrement). Also, the corresponding inhibitory neuron receives excitation, and starts inhibiting the initial memory position neuron. In Fig. 10.13 the sensor undergoes a few rotations both clockwise and counterclockwise, complete and half turns. Sensor's output pulse frequencies are plotted above, for the three gyroscope's axes; below the corresponding neurons' firing rate is shown, as a mean between the regular and irregular one for the preferred and non-preferred direction.

10.6 Discussion and Conclusions

We presented a hardware setup that implements a model of the vestibular system studied in primates. Due to the high degree of similarity between the vestibular system in non-human primates and humans, it is plausible to extend these results also to humans (see for example recent results from [Todd et al., 2008]), which is useful for applications in neuroprosthetics. In particular, our results faithfully reproduce the primary afferent neurons response of the vestibular system, which is similar for both the two vestibular

organs present in non-human primates and humans [Angelaki and Dickman, 2000]. Within this context, the system presented can be used as a basic research tool for studying the computational and physiological properties of real vestibular systems, and for investigating the neural mechanisms underlying more complex levels of perception which are still not fully understood. For example, the way we recognize gravity from linear acceleration (walking forward or tilting our head backward is indistinguishable to primary otolith afferents) seems to be related with more central brain areas which take advantages from both sensory responses [Angelaki and Cullen, 2008]. Biological nervous systems process signals using analog, asynchronous, and parallel architectures, comprising in-homogeneous and unreliable hardware. They adopt computing paradigms that are completely different from those of state-of-the-art digital processing systems. Although general purpose computing devices (by definition) could be used and programmed to reproduce specific features of real vestibular systems, the study of their inner workings, dynamics, internal state and signal representations, and failures and limitations, can provide only limited insight into the mechanisms underlying the behavior of the real system they simulate. We argue that the mixed-signal approach proposed here, to implement a physical model of vestibular signal processing in an embodied artificial behaving system, by means of analog subthreshold VLSI, can shed light on the principles of neural computation used in the real vestibular system. This is supported by the fact that the physics of the transistors used in the silicon neuron and synapse circuits are analogous to the physics of proteic channels in real neurons and synapses [Mead, 1990]; the subthreshold analog VLSI circuits, like the neural circuits they emulate, are in-homogeneous, affected by noise, and have low/limited resolution; neuromorphic computing architectures, like their biological counterparts, use parallel, distributed, and collective computational paradigms. In addition, by building a physical model of the vestibular system that faithfully reproduces the responses of the real system's neurons could provide important insights into the properties of biological vestibular systems under, for example, extreme conditions (e.g., with high velocities, accelerations or different gravity conditions) where animal experiments would be difficult or unethical. The constraints imposed on the implementation by the architecture of the sensor's processing unit have been met with respect to the required timing stability of the output signal (e.g., see Fig. 10.9). A commercial IMU sensor has been used, thus allowing the system description to be focused only on the neuromorphic hardware implementation. By following the parallelism with the biological system, the IMU sensor can be viewed as the current injection into the primary afferents. In order to accurately drive the silicon neurons, higher priority has been given to the IMU timer interrupt, and a lower one to the data acquisition. The use of a commercial IMU equipped with an ARM processor, which has only been used to encode the sensor's outputs in spikes, has

lead to a fast development of the full system. Although the system's RMS error (e.g., see Fig. 10.11) is very high if compared to that of systems used in conventional robotic applications, our results are comparable with those obtained from detailed biological models, and in turn, with real biological data. We showed how the neuromorphic systems developed can be used to implement biologically inspired position detectors, such as the head-direction detector models of Fig. 10.2. Realizing a hardware implementation of an integrator model using spiking neural systems is a challenge that few other groups tackled in the past [Choudhary et al., 2012; Massoud and Horiuchi, 2009]. In the theoretical models of the head direction cell system as in [Hahnloser, 2003; Xie et al., 2002; Zhang, 1996] the computation is performed in the mean rate domain, with attractor states [Amit, 1992] that allow the storage of information in self-sustained memories. The integration operation is performed by distinct pools of neurons that receive as input the differential intake of angular velocities. Our hardware model is based on the same principles, although we used a single calibrated neuron with strong recurrent connections instead of a distributed attractor state. In [Massoud and Horiuchi, 2009] the authors realized a head-direction neuromorphic system by integrating angular velocities with stable bumps of activities (attractor states) with small populations of silicon neurons. However, in their implementation the input signal was simulated using a global excitatory input current that was used to mimic rotation of the head, in the left or right direction.

The work proposed in [Choudhary et al., 2012] describes a more general integrator network where the authors use probabilistic weights controlled via a dedicated FPGA board, to achieve precise control, but which requires a large population of neurons. The integrator network that we propose can be implemented within the neuromorphic vestibular system framework, which requires only a small number silicon neurons and can be implemented using a very small VLSI real-estate (e.g., integrated in a few square millimeters area, with a conservative CMOS technology). In addition, this integrator network can be easily extended within the same framework to take into account also accelerations, in addition to velocities, allowing the system to keep track of both position and velocity with a resolution proportional to the number of neurons used. By replacing the commercial IMU unit with custom low-power MEMS gyroscopic sensors, such as those recently proposed in [Andreou et al., 2013], it will be possible to implement a compact low-power integrated system with interesting features both for robotic applications (e.g., in humanoid robots) and prosthetics.

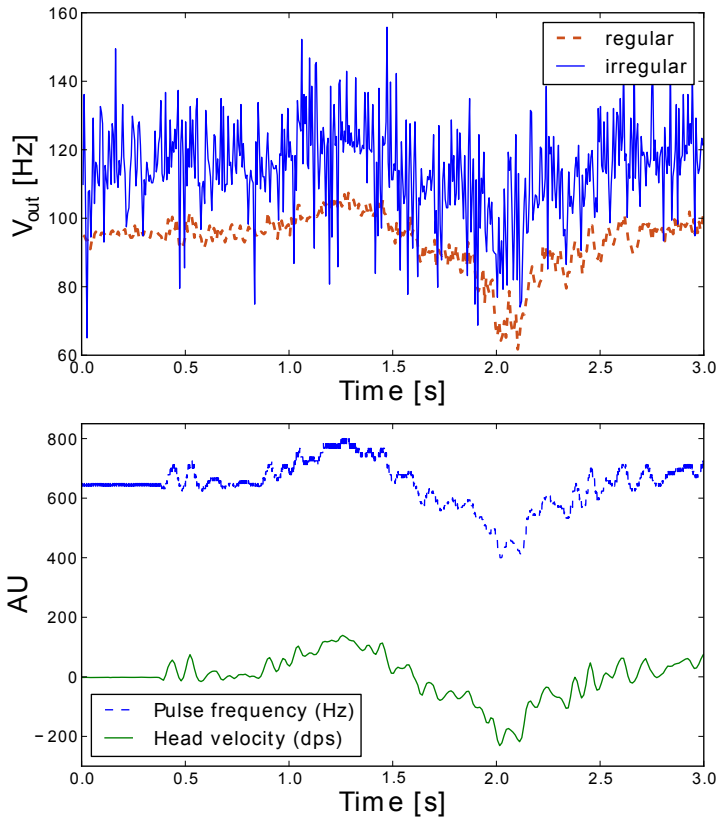


Figure 10.9: **Vestibular system input and output:** up) The output signal of the IMU unit (both angular velocity and associated PDM frequency). down) The corresponding neuron's mean firing rates in response to those inputs. Analogous plots have been measured for the Y and Z axes (Fig. 10.5).

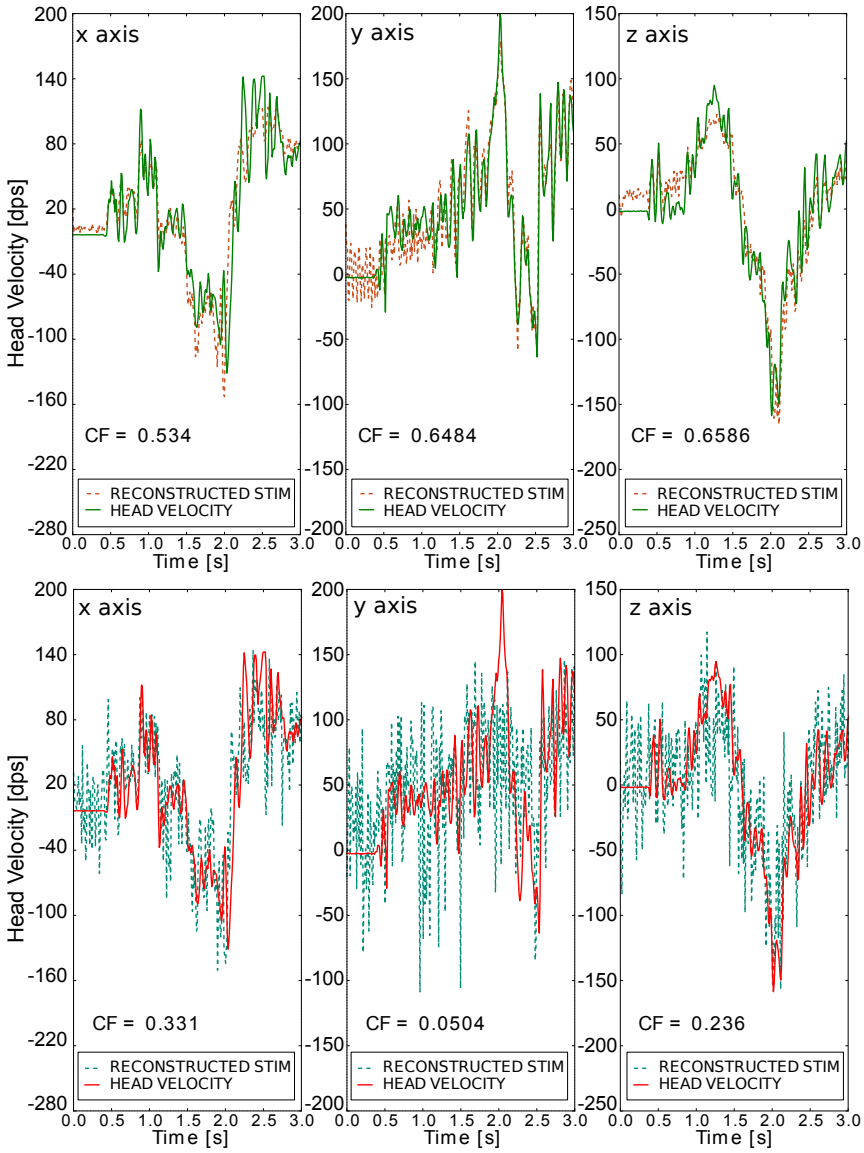


Figure 10.10: **Stimulus reconstruction from regular and irregular afferents in response to a broadband stimulus** (a) Regular neurons (b) Irregular neurons.

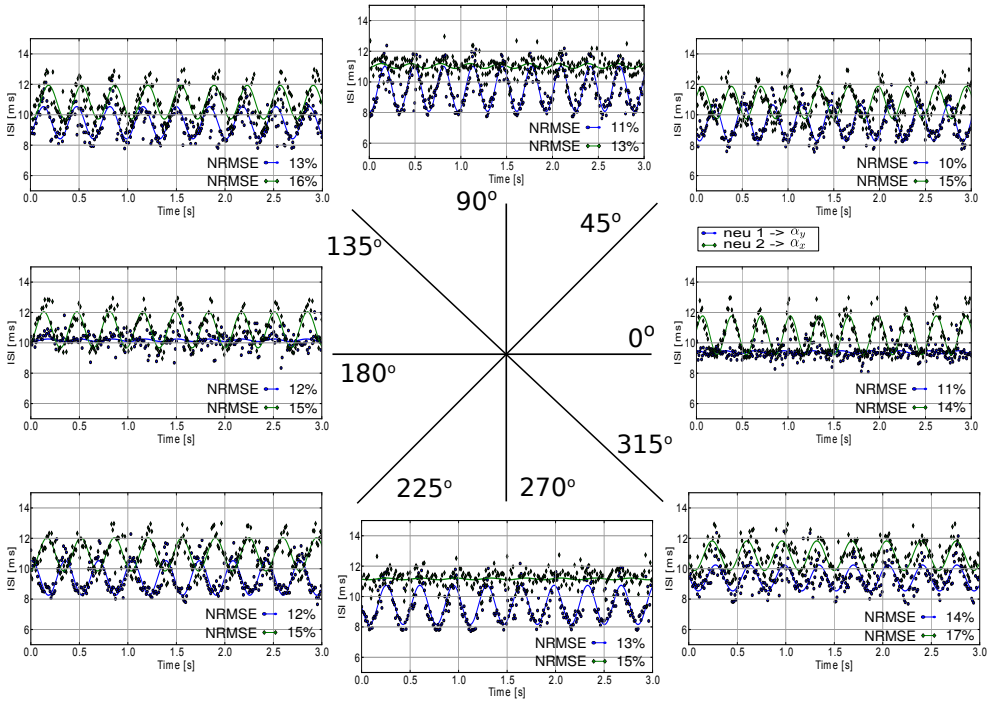


Figure 10.11: **Neurons' responses.** Each plot shows the inter-spike-interval of two regular neurons, for different motions on the horizontal plane. Stimulus orientations of 0° and 180° correspond to lateral motion, whereas 90° corresponds to fore-aft motion. The accelerations impressed at the sensor are sinusoids with an amplitude of about $\pm 0.57\,\mathrm{G}$. ($\mathrm{G} = 9.81\,\mathrm{m/s^2}$). All the plots are aligned with the start of the stimulation and continuous lines highlight the best-fit sine function. Note that the neurons respond correctly to both their preferred stimulus orientation by modulating their firing rate, and to the phase of the stimulation.

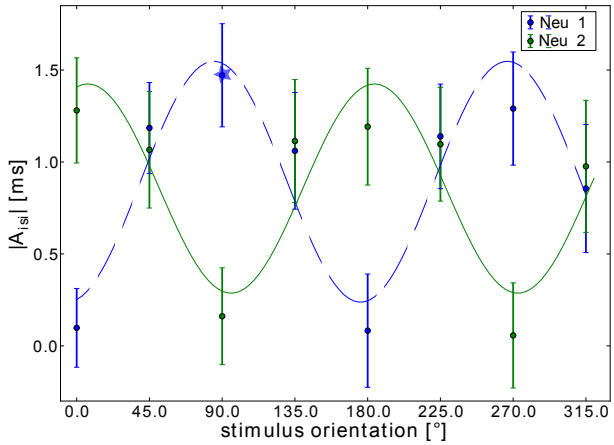


Figure 10.12: **Inter-Spike-Interval Amplitude in function of stimulus orientation for two regular afferent neurons.** The data are obtained from the fits visible in Fig. 10.11. Amplitudes are maximal for the preferred stimulus orientations $90°, 270°$ and $0°, 180°$ respectively. Error bars represent the Root-Mean-Squared Error between the fitted sine wave and the data points visible in figure 10.11.

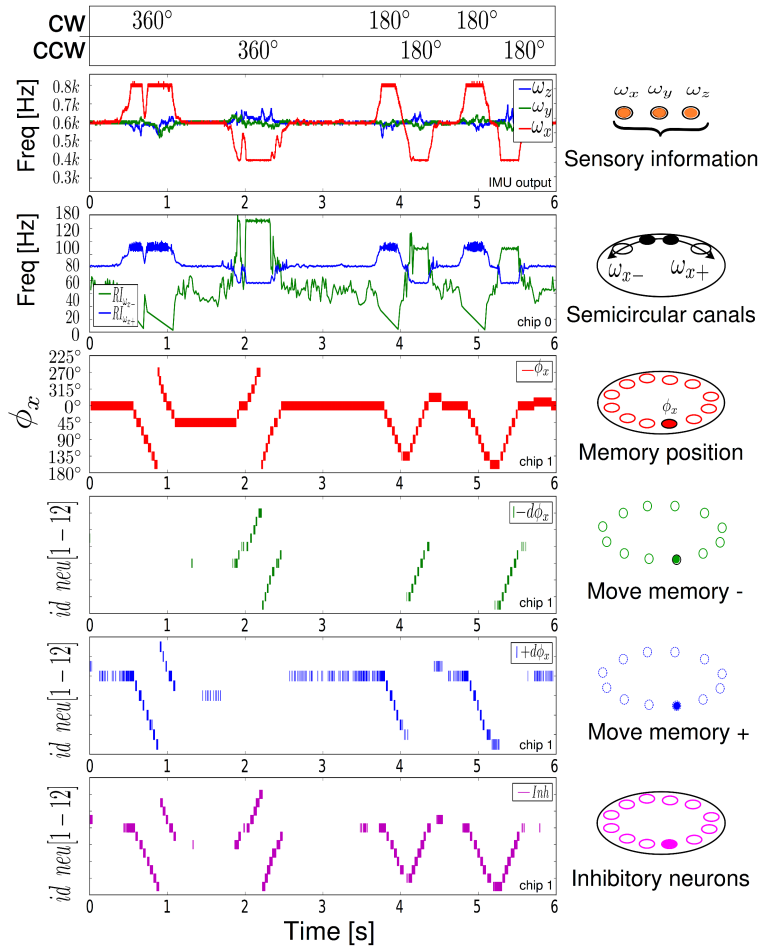


Figure 10.13: **Integrator behavior.** The top table indicates sensor's rotation value along the x-axis. Rotations are clockwise (cw) or counterclockwise (ccw) as indicated by the row. The first plot shows the IMU angular velocities output along the three axis (x, y, z). The second plot (chip 0) shows the mean firing rate for the two regular and irregular neurons encoding directions of positive and negative rotations. The third plot (chip1) is a raster plot of the neurons in the memory layer responsible for storing the current angular position. Decrement, increment and inhibitory neurons (chip 1) are shown in the three remaining plots. The noise in the sensor's output at $t \approx 0.5, 2s$, which appears in the top plot, is caused by the friction of the cables.

CHAPTER 11

Discussion

Understanding the computational primitives and the architectural principles of the human brain can be extremely important for future Information and Communication Technologies (ICT). For this reason, many laboratories and private companies are studying alternative approaches to standard von Neumann computing architectures. Many of these approaches are aimed at the realization of compact, energy-efficient computational devices that map the style of computation of the brain into artificial systems. Neuromorphic microelectronic systems represent one promising approach in this respect [Boahen et al., 2015; Markram, 2012; McQuinn et al., 2013; Mead, 1990; Modha et al., 2011] as they implement event-based neural inspired distributed models of neural computation. Neuromorphic systems can be considered part of a new generation of technologies that have the potential to increase performances, and decrease power consumption Hasler and Marr [2013] while instantiating cognitive behaviors [Neftci et al., 2013].

11.1 Artificial intelligent systems

Today's notion of artificial intelligent systems is mostly associated with super computers capable of beating the world champion in chess [Hsu, 1999], or beating humans in live TV quizzes [Ferrucci et al., 2010; Markoff, 2011]. These systems are somehow similar to "HAL9000" in the movie "2001: A Space Odyssey"; they all are competent of executing a specific task such as playing chess, playing quiz games on general knowledge, and controlling the system of the spaceship "Discovery One". This fast progress of machine intelligence has been supported by conventional general-purpose computers that have

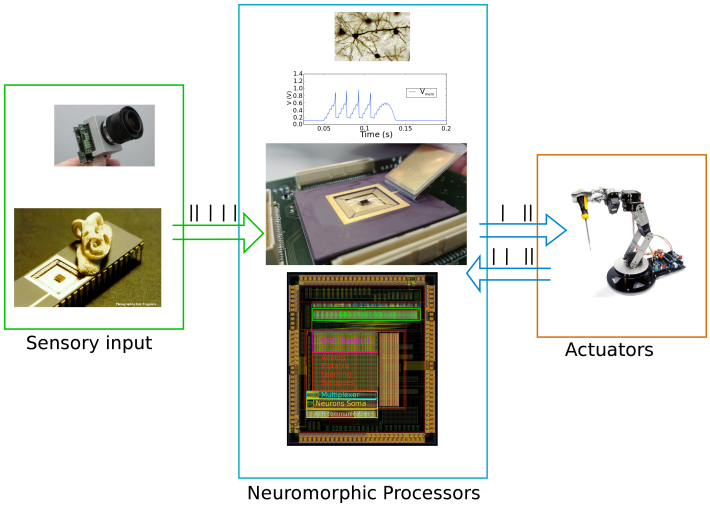


Figure 11.1: Examples of neuromorphic computing systems

become faster, more powerful, and user friendly. All this has been largely financed by the gigantic electronic industry that has grown at an impressive pace. However, technical limits imposed by the von Neumann bottleneck problem and the physical limits of VLSI integration are starting to emerge [Frank et al., 2001; Likharev, 2003]. In recent years, massive amounts of data have been collected thanks to the digital revolution, and trying to make sense of all this data with standard algorithmic computers is becoming technically very challenging. For these reasons, and thanks to new theoretical understandings [Hinton and Salakhutdinov, 2006], an entire field of artificial neural network applications named *deep-learning* has become very popular. The field aims at processing the information organized in large databases for extracting relationships in the data at multiple levels of abstractions. The approach is based on training large networks with the backpropagation learning algorithm to indicate which of the internal connections should change. These connections are used to compute representations in each part of the network based on the previous part of the network (i.e. layer). This approach has recently been demonstrated to outperform state-of-the-art approaches in various applications that range from image classifiers, to language processing, to speech recognition systems [Le Cun et al., 2015]. Even if the deep-learning approach is extremely useful, its success can still be related to the increased computational power, and to the large amount of data that have become available [Cireşan et al., 2010]. The back-propagation algorithm is not new, see [Bryson et al., 1963; Bryson, 1975; Rumelhart et al., 1986], but only recently it is being fully appreciated. It is not only the increase in computational power that is making deep-learning interesting, but also the fact that now it is possible to

use unsupervised methods for learning features [Hinton and Sejnowski, 1999]. This has greatly sped up the progress by eliminating the human role of engineering good features. Learning the features is usually done thanks to the use of powerful Graphical Processing Unit (GPU)s to process the data [Chetlur et al., 2014]. In particular, learning good features requires the use of massive amounts of data, resulting in a very costly procedure in terms of power and time. In this direction, implementations of spiking neural networks in neuromorphic hardware offer promising solutions for running deep networks on low-power hardware [Stromatias et al., 2015]. There are however a series of technical challenges that must be addressed in order to be able to run deep-networks in neurmorphic hardware. These challenges are related to hardware limitations such as noise, limited weights precision, as well as the need to deal with real-world unpredictable stimuli. Work in this direction has been started both at a theoretical level [Diehl et al., 2015; Muller and Indiveri, 2015], as well as in physical systems implementations [Neil and Liu, 2014; Stromatias et al., 2015]. Alternative paths to intelligent machines rely on the notion of embodiment, in which artificial systems are physical entities that interact within the real-world, something like "R2-D2", the droid companion in "Star Wars" movies. An example of such path is represented by robotic systems, in which it is essential some interaction between the real-world and the internal state of the robot. Neuromorphic engineering also plays a role in this direction as robotic systems require compact, low-power, and low-latency computing elements. Many demonstrations of the neuromorphic approach in robotic systems are available to showcase this form of intelligent machines [Conradt et al., 2009; Corradi and Giulioni, 2012; Delbruck and Lang, 2013; O'Connor et al., 2013]. In addition, neuromorphic engineering is well suited for the exploitation of new materials and devices that can be used to instantiate neural-like computing elements. New materials as memristors [Indiveri et al., 2013; Prezioso et al., 2015], carbon-nanotubes [Zhao et al., 2010], and nanowire based transistors [Türel and Likharev, 2003] are currently analyzed as valid options that could enable a much better scaling thanks to the combined use of hybrid semiconductor/nanodevice-integrated circuits. The neuromorphic engineering approach has a clear advantage when dealing with such devices compared to the standard digital logic approaches to computation. These advantages come from the principles of computations and models that have been developed within the neuromorphic community. Neural models of computation are used to deal with mismatch, noise and heterogeneous computational substrates as they are inspired by the similar computational primitives of neurobiological systems. Fruitful collaborations between different fields are necessary in order to push further the development of intelligent machines.

11.2 Small-scale reconfigurable neuromorphic hardware

There are good reasons to develop neuromorphic devices composed of relatively small number of units (in the order of hundreds of neurons and thousands of synapses). Small scale networks might be good candidate for the representation of computational primitives which are generic and reusable, and their composition could give rise to the cortical sheet [Douglas and Martin, 2004]. These components can be seen as small recurrent networks in which excitatory and inhibitory feedback loops are used to process the information, while shaping network dynamics. These small-scale networks are also used as fundamental components in neuromorphic systems, much like registers and adders represent the basic components of digital general-purpose computers. Second, the neuromorphic systems are designed using a mixed signal analog/digital approach. This poses a series of technical challenges as we must deal with: heterogeneities in the device, mismatch, cross-talk between analog circuits and digital logic blocks, and the spiking nature of our devices requires the realization of massively parallel asynchronous communication infrastructure. For these type of problems automatic tools do not yet exist. We must therefore address all these challenges and come up with ad hoc solutions [Liu, 2015]. These challenges must be solved if we want to develop neural-like devices that enable the study of computational primitives and that can represent a valid substrate for emulating larger spiking neural networks.

11.3 Benefits and drawbacks of asynchronous logic circuits

Asynchronous logic circuits represent an appealing way for encoding information in spiking neural networks. The analogy with biological neural nets is that an event represents an action potential (a spike). This analogy has several implications in the structure and in the way neuromorphic devices communicate. In asynchronous logic systems there is no notion of common or discrete time. The components of an asynchronous system perform the necessary operations of communication, and synchronization, using handshaking mechanisms that are activated only at the time in which events are produced. In a large system this give rise to communication of information in a grained way, in time and space. This grained communication is beneficial in term of power, as only the active part of the system would be required to operate. In addition, the static power of asynchronous logic circuits is usually extremely low [Van Berkel et al., 1994]. The fact that there is no global clock in the system naturally eliminates problems of clock distribution, as well as clock skew problems. These problems represent hard limits on the speed of standard clocked systems. Therefore asynchronous design could also result in an increase in operational speed [Martin et al., 1997] as the actual speed is limited by neighbours latencies instead

than worst-case scenarios. In addition, asynchronous design appears to be more robust towards fabrication process variations [Nielsen et al., 1994]. This increased robustness is caused by the fact that synchronization is achieved by means of matched delays. All these features allows for a modular, and composite design in which handshaking interfaces take care of local timing issues. Even if there are many benefits in using asynchronous logic while realizing communication between and within neuromorphic devices, there are also some drawbacks. These drawbacks are mainly caused by the overhead costs that handshaking mechanisms bring in. These overheads costs are represented by an increase in silicon area. Whether the use of asynchronous techniques is beneficial or not is of course a trade-off between its overhead cost and the architectural benefits.

11.4 Summary and Relevance

This thesis embraces different aspects of the neuromorphic approach to computation. It starts with the description of basic circuits for modeling network elements: neurons and synapses. In Chapter 3 I described and applied a method, based on dynamical system theory, useful to find relevant parameter ranges in the space of all possible parameters, for reproducing bio-realistic behaviors of the silicon neuron. Then I described different neuromophic devices that have been developed (Chapters 4, 5), and that have been used to efficiently map neuroscientific models of neural computation onto neuromorphic hardware. These devices represent a step toward more compact and programmable event-based neuromorphic processors. In these systems, and in contrast to all von Neumann architectures, there is no separation between memory elements and computational units. This separation supports the scalability of neuromorphic architectures by avoiding the bottleneck communication problem[Backus, 1978]. In Chapters 6, 7, and 8 I demonstrated, in different computational settings, that it is possible to control feed-forward as well as recurrent VLSI networks to achieve interesting computation. Some of the dynamical patterns of activity observed in the cortex of behaving animals have been emulated in these devices, demonstrating that these neuromorphic systems exploits similar computational principles to real neuro-biological systems. Applications in which neural-like computing devices can be beneficial are found in robotics (in particular in bio-inspired robotic systems), but also in medical applications as BMI, and prostheses, and implantable devices. From this perspective, Chapters 9, and 10 demonstrated results that can be readily used to develop such neural-inspired BMI devices and bio-mimetic prosthetic controllers.

11.5 Outlook and Conclusions

Studying models of neural computation and emulating them in silicon substrates can be beneficial from both a technological and theoretical point of view. It is also important to start applying these computational principles and the resulting technologies in the development of embedded devices for practical applications in real-world scenarios. This work provides some important building blocks, tools, and models to make progress in this direction. More generally, this work provides a route towards a new generation of computing technologies that are alternative to standard computer architectures, and might enable cognitive and intelligent computing. In addition, we developed silicon devices and systems that go towards a new generation of BMI, and prosthetic devices that apply neuromorphic principles in their structural organization, and in their computational basis.

Acknowledgments

A considerable amount of people have had an influence in the work of this thesis from multiple perspectives. Here, I would like to acknowledge all of them.

I am extremely thankful to my supervisor Prof. Giacomo Indiveri who has always given me the freedom to pursue various projects without objections; I really admire his temperament and his open mindset. I am very much thankful to Prof. Tobi Delbruck for having unconsciously taught me, among many other things, his smooth attitude.

All of my colleagues that are/have been part of the Neuromorphic Cognitive Systems Group, in Zurich or the Complex Systems Laboraty Group, in Rome, have had a big impact on the work of this thesis. In particular Ning Qiao, Hesham Mostafa, Marc Osswald, Fabio Stefanini, Sabed Moradi, Timoleon Moraitis, Sim Bamford, Massimiliano Giulioni, Honzghi You, and Sadique Sheik; they all have been involved at some stage of the journey, and in different ways.

I am immensely thankful to my beloved Gabriella Hàzi for constantly reminding me that in life there is much more than just science.

Finally, my deep and sincere gratitude to my family for their continuous love. This journey would have not been possible without their encouragements to seek my passions.

Bibliography

L.F. Abbott and S.B. Nelson. Synaptic plasticity: taming the beast. *Nature Neuroscience*, 3:1178–1183, November 2000.

S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.

José M Amigó, Janusz Szczepański, Elek Wajnryb, and Maria V Sanchez-Vives. Estimating the entropy rate of spike trains via lempel-ziv complexity. *Neural Computation*, 16(4):717–736, 2004.

Daniel J. Amit, Hanoch Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, 55:1530–1533, Sep 1985. doi: 10.1103/PhysRevLett.55.1530. URL `http://link.aps.org/doi/10.1103/PhysRevLett.55.1530`.

D.J. Amit. *Modeling brain function: The world of attractor neural networks*. Cambridge University Press, 1992.

D.J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex*, 7:237–252, 1997.

D.J. Amit and G. Mongillo. Spike-driven synaptic dynamics generating working memory states. *Neural Computation*, 15(3):565–596, 2003.

Rajagopal Ananthanarayanan and Dharmendra S Modha. Anatomy of a cortical simulator. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, page 3. ACM, 2007.

Charalambos M Andreou, Yiannis Pahitas, Evdokia Pilavaki, and Julius Georgiou. Bio-mimetic gyroscopic sensor for vestibular prostheses. In *Biomedical Circuits and Systems Conference, (BioCAS), 2013*, pages 17–20. IEEE, 2013.

Dora E Angelaki and Kathleen E Cullen. Vestibular system: the many facets of a multimodal sense. *Annu. Rev. Neurosci.*, 31:125–150, 2008.

Dora E Angelaki and J D Dickman. Spatiotemporal processing of linear acceleration: primary afferent and central vestibular neuron responses. *Journal of neurophysiology*, 84(4):2113–2132, October 2000.

J.V. Arthur, P. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, A. Chandra, S. Esser, N. Imam, W. Risk, D.B.D. Rubin, R. Manohar, and D.S. Modha. Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. In *International Joint Conference on Neural Networks, IJCNN 2012*, pages 1946–1953. IEEE, Jun 2012. doi: 10.1109/IJCNN.2012.6252637.

J.N.Y. Aziz, Karim Abdelhalim, R. Shulyzki, R. Genov, B.L. Bardakjian, M. Derchansky, D. Serletis, and P.L. Carlen. 256-channel neural recording and delta compression microsystem with 3d electrodes. *IEEE Journal of Solid-State Circuits*, 44(3):995–1005, March 2009. ISSN 0018-9200. doi: 10.1109/JSSC.2008.2010997.

J. Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, 1978. doi: 10.1145/359576.359579. URL `http://doi.acm.org/10.1145/359576.359579`.

D. Badoni, M. Giulioni, V. Dante, and P. Del Giudice. An aVLSI recurrent network of spiking neurons with reconfigurable and plastic synapses. In *International Symposium on Circuits and Systems, (ISCAS), 2006*, pages 1227–1230. IEEE, May 2006.

S.-A. Bamford, A.-F. Murray, and D.-J. Willshaw. Silicon synapses self-correct for both mismatch and design inhomogeneities. *Electronics letters*, 18:360–361, 2013. ISSN 0013-5194. doi: 10.1049/el.2012.0257.

Simeon A Bamford, Roni Hogri, Andrea Giovannucci, Aryeh H Taub, Ivan Herreros, Paul FMJ Verschure, Matti Mintz, and Paolo Del Giudice. A vlsi field-programmable mixed-signal array to perform neural signal processing and neural modeling in a prosthetic system. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(4):455–467, 2012.

O. Barak, M. Rigotti, and S. Fusi. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *The Journal of Neuroscience*, 33(9):3844–3856, 2013.

D Barsakcioglu, Y Liu, P Bhunjun, J Navajas, A Eftekhar, A Jackson, Quiroga R Quian, and TG Constandinou. An analogue front-end model for developing neural spike sorting systems. *IEEE Transactions on Biomedical Circuits and Systems*, 8:216–227, 2014. doi: 10.1109/TBCAS.2014.2313087. URL `http://dx.doi.org/10.1109/TBCAS.2014.2313087`.

C. Bartolozzi and G. Indiveri. A selective attention multi–chip system with dynamic synapses and spiking neurons. In B. Schölkopf, J.C. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 19, pages 113–120, Cambridge, MA, USA, Dec 2007a. Neural Information Processing Systems Foundation, MIT Press. URL `http://ncs.ethz.ch/pubs/pdf/Bartolozzi_Indiveri06c.pdf`.

C. Bartolozzi and G. Indiveri. Synaptic dynamics in analog VLSI. *Neural Computation*, 19(10):2581–2603, Oct 2007b. doi: 10.1162/neco.2007.19.10.2581. URL `http://ncs.ethz.ch/pubs/pdf/Bartolozzi_Indiveri07.pdf`.

B. Belhadj, J. Tomas, O. Malot, G. N'Kaoua, Y. Bornat, and S. Renaud. Fpga-based architecture for real-time synaptic plasticity computation. In *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, pages 93–96, Aug 2008.

Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R Chandrasekaran, J Bussat, R Alvarez-Icaza, JV Arthur, PA Merolla, and K Boahen. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proceedings of the IEEE*, 102(5):699–716, 2014.

A. Berthoz. *The Brain's Sense of Movement, Perspectives in Cognitive Neuroscience*. Harvard University Press, 2002.

G. Billings and M.C.W. van Rossum. Memory retention and spike-timing-dependent plasticity. *Journal of Neurophysiology*, 101(6):2775–2788, Mar 2009. doi: 10.1152/jn.91007.2008.

K.A. Boahen. Point-to-point connectivity between neuromorphic chips using address-events. *IEEE Transactions on Circuits and Systems II*, 47(5):416–34, 2000.

Kwabena Boahen, Rajit Manohar, and Chris Eliasmith. Neuromorphics: Programmable analog computation through probabilistic digital communication, 2015.

M. Boegerhausen, P. Suter, and S.-C. Liu. Modeling short-term synaptic depression in silicon. *Neural Computation*, 15(2):331–348, Feb 2003.

T. Borghi, A. Bonfanti, G. Zambra, R. Gusmeroli, A.L. Lacaita, A.S. Spinelli, and G. Baranauskas. An integrated low-noise multichannel system for neural signals amplification. *33rd European Solid State Circuits Conference, 2007. ESSCIRC*, pages 456–459, Sept. 2007. ISSN 1930-8833. doi: 10.1109/ESSCIRC.2007.4430341.

Alexander Borst and Juergen Haag. Effects of mean firing on neural information rate. *Journal of computational neuroscience*, 10(2):213–221, 2001.

J. Brader, W. Senn, and S. Fusi. Learning real world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Computation*, 19:2881–2912, 2007.

Jochen Braun and Maurizio Mattia. Attractors and noise: twin drivers of decisions and multistability. *Neuroimage*, 52(3):740–751, 2010.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996. ISSN 0885-6125. doi: 10.1007/BF00058655.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

R. Brette and W. Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of Neurophysiology*, 94:3637–3642, 2005.

S Brink, S Nease, and P Hasler. Computing with networks of spiking neurons on a biophysically motivated floating-gate based neuromorphic integrated circuit. *Neural networks: the official journal of the International Neural Network Society*, 45:39, 2013.

Daniel Bruederle, Mihai Petrovici, Bernhard Vogginger, Matthias Ehrlich, and et al. A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biological Cybernetics*, 104:263–296, 2011. doi: 10.1007/s00422-011-0435-9.

Arthur E Bryson, Walter F Denham, and Stewart E Dreyfus. Optimal programming problems with inequality constraints. *AIAA journal*, 1(11):2544–2550, 1963.

Arthur Earl Bryson. *Applied optimal control: optimization, estimation and control*. CRC Press, 1975.

A. Cassidy and A.G. Andreou. Dynamical digital silicon neurons. In *Biomedical Circuits and Systems Conference, (BioCAS), 2008*, pages 289–292. IEEE, Nov. 2008. doi: 10.1109/BIOCAS.2008.4696931.

A.S. Cassidy, J. Georgiou, and A.G. Andreou. Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization. *Neural Networks*, 2013. doi: 10.1016/j.neunet.2013.05.011. URL `http://www.sciencedirect.com/science/article/pii/S0893608013001597`.

C.T. Charles. Wireless data links for biomedical implants: Current research and future directions. In *Biomedical Circuits and Systems Conference, (BioCAS), 2007*, pages 13–16. IEEE, Nov. 2007. doi: 10.1109/BIOCAS.2007.4463297.

Yi Chen, Anirban Basu, Lei Liu, Xiaodan Zou, Ramamoorthy Rajkumar, Gavin Stewart Dawe, and Minkyu Je. A digitally assisted, signal folding neural recording amplifier. *Biomedical Circuits and Systems, IEEE Transactions on*, 8(4):528–542, 2014.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

K. Cheung, S.R. Schultz, and W. Luk. A large-scale spiking neural network accelerator for FPGA systems. In *Artificial Neural Networks and Machine Learning–ICANN 2012*, pages 113–120. Springer, 2012.

E. Chicca. A VLSI neuromorphic device with 128 neurons and 3000 synapses: area optimization and design. Master's thesis, University of Rome 1, "La Sapienza", 1999. In Italian.

E. Chicca, F. Stefanini, C. Bartolozzi, and G. Indiveri. Neuromorphic electronic circuits for building autonomous cognitive systems. *Proceedings of the IEEE*, 102(9):1367–1388, Sep 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2313954. URL `http://ncs.ethz.ch/pubs/pdf/Chicca_etal14.pdf`.

S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen. Silicon neurons that compute. In A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, editors, *Artificial Neural Networks and Machine Learning – ICANN 2012*, volume 7552 of *Lecture Notes in Computer Science*, pages 121–128. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-33268-5. doi: 10.1007/978-3-642-33269-2_16.

D.C. Cireşan, U. Meier, L.M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural computation*, 22(12):3207–3220, 2010.

J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R.J. Douglas, and T. Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors. In *International Symposium on Circuits and Systems, (ISCAS), 2009*, pages 781–784. IEEE, May 2009. doi: 10.1109/ISCAS.2009.5117867.

T.G. Constandinou, J. Georgiou, and C. Toumazou. A partial-current-steering biphasic stimulation driver for vestibular prostheses. *IEEE Transactions on Biomedical Circuits and Systems*, 2(2):106–113, 2008a.

T.G. Constandinou, J. Georgiou, and C. Toumazou. Towards an integrated, fully-implantable vestibular prosthesis for balance restoration. *Advances in Science and Technology*, 57:210–215, 2008b.

F. Corradi and G. Indiveri. A neuromorphic event-based neural recording system for smart brain-machine-interfaces. *Biomedical Circuits and Systems, IEEE Transactions on*, 2015. (submitted).

F. Corradi, D. Bontrager, and G. Indiveri. Toward neuromorphic intelligent brain-machine interfaces: An event-based neural recording and processing system. In *Biomedical Circuits and Systems Conference (BioCAS)*, pages 584–587. IEEE, Oct 2014a. doi: 10.1109/BioCAS.2014.6981793. URL `http://ncs.ethz.ch/pubs/pdf/Corradi_etal14c.pdf`.

F. Corradi, C. Eliasmith, and G. Indiveri. Mapping arbitrary mathematical functions and dynamical systems to neuromorphic vlsi circuits for spike-based neural computation. In *International Symposium on Circuits and Systems, (ISCAS), 2014*, pages 269–272. IEEE, 2014b. URL `http://ncs.ethz.ch/pubs/pdf/Corradi_etal14.pdf`.

F. Corradi, D. Zambrano, M. Raglianti, G. Passetti, C. Laschi, and G. Indiveri. Towards a neuromorphic vestibular system. *Biomedical Circuits and Systems, IEEE Transactions on*, 8(5):669–680, Oct 2014c. ISSN 1932-4545. doi: 10.1109/TBCAS.2014.2358493. URL `http://ncs.ethz.ch/pubs/pdf/Corradi_etal14b.pdf`.

F. Corradi, H. You, M. Giulioni, and G. Indiveri. Decision making and perceptual bistability in spike-based neuromorphic vlsi systems. In *International Symposium on Circuits and Systems, (ISCAS), 2015*. IEEE, 2015. URL `http://ncs.ethz.ch/pubs/pdf/Corradi_etal15.pdf`.

Federico Corradi and Massimiliano Giulioni. Learning to recognize visual stimuli in neuromorphic vlsi. In *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, pages 90–90. IEEE, 2012.

J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco. A spatial contrast retina with on-chip calibration for neuromorphic spike-based AER vision systems. *IEEE Transactions on Circuits and Systems I*, 54(7):1444–1458, 2007. doi: 10.1109/TCSI.2007.900179.

K. Cullen and S. Sadeghi. Vestibular system, 2008. URL `www.scholarpedia.org/article/Vestibular_system`.

Nuno Maçarico da Costa and Kevan AC Martin. Whose cortical column would that be? *Frontiers in neuroanatomy*, 4, 2010.

P. Dario, M.C. Carrozza, E. Guglielmelli, C. Laschi, A. Menciassi, S. Micera, and F. Vecchi. Robotics as a future and emerging technology biomimetics, cybernetics, and neuro-robotics in european projects. *IEEE Robotics & Automation Magazine*, 12(2):29–45, June 2005.

Paul Dean, John Porrill, and James V Stone. Decorrelation control by the cerebellum achieves oculomotor plant compensation in simulated vestibulo-ocular reflex. *Proceedings of the Royal Society B: Biological Sciences*, 269(1503):1895–1904, September 2002.

G. Deco and E. Rolls. Object-based visual neglect: a computational hypothesis. *European Journal of Neuroscience*, 16:1994–2000, 2002.

G. Deco, E.T. Rolls, and B. Horwitz. 'What' and 'where' in visual working memory: a computational neurodynamical perspective for integrating fMRI and single-neuron data. *Journal of Cognitive Neuroscience*, 16:683–701, 2004.

S.R. Deiss, R.J. Douglas, and A.M. Whatley. A pulse-coded communications infrastructure for neuromorphic systems. In W. Maass and C.M. Bishop, editors, *Pulsed Neural Networks*, chapter 6, pages 157–78. MIT Press, 1998.

P. Del Giudice, S. Fusi, and M. Mattia. Modeling the formation of working memory with networks of integrate-and-fire neurons connected by plastic synapses. *Journal of Physiology Paris 97*, pages 659–681, 2003.

T. Delbruck and M. Lang. Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor. *Frontiers in Neuroscience*, 7(223), 2013. ISSN 1662-453X. doi: 10.3389/fnins.2013.00223. URL `http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2013.00223/abstract`.

T. Delbruck, R. Berner, P. Lichtsteiner, and C. Dualibe. 32-bit configurable bias current generator with sub-off-current capability. In *International Symposium on Circuits and Systems, (ISCAS), 2010*, pages 1647–1650, Paris, France, 2010a. IEEE, IEEE. doi: 10.1109/ISCAS.2010.5537475.

T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch. Activity-driven, event-based vision sensors. In *International Symposium on Circuits and Systems, (ISCAS), 2010*, pages 2426–2429, Paris, France, 2010b. IEEE. doi: 10.1109/ISCAS.2010.5537149.

T. Delbruck, V. Villanueva, and L. Longinotti. Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision. In *International Symposium on Circuits and Systems, (ISCAS), 2014*, pages 2636–2639, Melbourne, Australia, 2014. IEEE, IEEE.

Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. 2015.

R.J. Douglas and K.A.C. Martin. Neural circuits of the neocortex. *Annual Review of Neuroscience*, 27: 419–51, 2004.

R.J. Douglas, K.A.C. Martin, and D. Whitteridge. A canonical microcircuit for neocortex. *Neural Computation*, 1:480–488, 1989.

R.J. Douglas, M.A. Mahowald, and C. Mead. Neuromorphic analogue VLSI. *Annu. Rev. Neurosci.*, 18: 255–281, 1995.

J-P Eckmann and David Ruelle. Ergodic theory of chaos and strange attractors. *Reviews of modern physics*, 57(3):617, 1985.

Jean-Pierre Eckmann, S Oliffson Kamphorst, and David Ruelle. Recurrence plots of dynamical systems. *Europhys. Lett*, 4(9):973–977, 1987.

V. Ekanayake and R. Manohar. Asynchronous DRAM design and synthesis. In *Ninth International Symposium on Asynchronous Circuits and Systems*, pages 174–183, Vancouver, BC, May 2003. IEEE.

C Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation and dynamics in neurobiological systems*. MIT Press, Cambridge, MA, 2003.

C. Eliasmith, T.C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012. doi: 10.1126/science.1225266. URL `http://www.sciencemag.org/content/338/6111/1202.abstract`.

Kai Olav Ellefsen, Jean-Baptiste Mouret, Jeff Clune, and Josh C Bongard. Neural modularity helps organisms evolve to learn new skills without forgetting old skills. *PLoS Comput Biol*, 11(4):e1004128, 2015.

A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature Reviews Neuroscience*, 9(4):292–303, 2008.

C. Farabet, C. Couprie, L. Najman, and Y. Le Cun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, Aug 2013. doi: 10.1109/TPAMI.2012.231.

E. Farquhar and P. Hasler. A bio-physically inspired silicon neuron. *IEEE Transactions on Circuits and Systems*, 52(3):477–488, March 2005.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

E. Franchi, E. Falotico, D. Zambrano, G. Muscolo, L. Marazzato, P. Dario, and C. Laschi. A comparison between two bio-inspired adaptive models of vestibulo-ocular reflex (vor) implemented on the icub robot. In *2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010)*, pages 251–256. IEEE, 2010.

David J Frank, Robert H Dennard, Edward Nowak, Paul M Solomon, Yuan Taur, and Hon-Sum Philip Wong. Device scaling limits of si mosfets and their application dependencies. *Proceedings of the IEEE*, 89(3):259–288, 2001.

Gene Frantz. Digital signal processor trends. *IEEE micro*, 20(6):52–59, 2000.

Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, 1986.

S.B. Furber, F. Galluppi, S. Temple, and L.A. Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, May 2014. ISSN 0018-9219. doi: 10.1109/JPROC.2014.2304638.

S. Fusi. Hebbian spike-driven synaptic plasticity for learning patterns of mean firing rates. *Biological Cybernetics*, 87:459–470, 2002.

S. Fusi. Spike-driven synaptic plasticity for learning correlated patterns of mean firing rates. *Rev Neurosci*, 14(1-2):73–84, 2003.

S. Fusi and L.F. Abbott. Limits on the memory storage capacity of bounded synapses. *Nature Neuroscience*, 10:485–493, 2007.

S. Fusi and M. Mattia. Collective behavior of networks with linear (VLSI) integrate and fire neurons. *Neural Computation*, 11:633–52, 1999.

S. Fusi, M. Annunziato, D. Badoni, A. Salamon, and D.J. Amit. Spike–driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural Computation*, 12:2227–58, 2000.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975, 2008.

Dileep George and Jeff Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS computational biology*, 5(10):e1000532, 2009.

Guido Gigante, Maurizio Mattia, Jochen Braun, and Paolo Del Giudice. Bistable perception modeled as competing stochastic integrations at two levels. *PLoS computational biology*, 5(7):e1000430, 2009.

M. Giulioni, P. Camilleri, V. Dante, D. Badoni, G. Indiveri, J. Braun, and P. Del Giudice. A VLSI network of spiking neurons with plastic fully configurable "stop-learning" synapses. In *International Conference on Electronics, Circuits, and Systems, ICECS 2008*, pages 678–681. IEEE, 2008. doi: 10.1109/ICECS.2008.4674944. URL `http://ncs.ethz.ch/pubs/pdf/Giulioni_etal08.pdf`.

M. Giulioni, M. Pannunzi, D. Badoni, V. Dante, and P. Del Giudice. Classification of correlated patterns with a configurable analog VLSI neural network of spiking neurons and self-regulating plastic synapses. *Neural Computation*, 21(11):3106–3129, 2009. doi: 10.1162/neco.2009.08-07-599.

M. Giulioni, P. Camilleri, M. Mattia, V. Dante, J. Braun, and P. Del Giudice. Robust working memory in an asynchronously spiking neural network realized in neuromorphic VLSI. *Frontiers in Neuroscience*, 5(149), 2012. ISSN 1662-453X. doi: 10.3389/fnins.2011.00149. URL `http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00149/abstract`.

M. Giulioni, F. Corradi, V. Dante, and P. Del Giudice. Real time unsupervised learning of visual stimuli in neuromorphic vlsi systems. *arXiv preprint arXiv:1506.05427*, 2015.

Paul W Glimcher. The neurobiology of visual-saccadic decision making. *Annual review of neuroscience*, 26(1):133–179, 2003.

Jay M Goldberg, C E Smith, and Cesar Fernández. Relation between discharge regularity and responses to externally applied galvanic currents in vestibular nerve afferents of the squirrel monkey. *Journal of neurophysiology*, 51(6):1236–1256, June 1984.

D. Goodman and R. Brette. Brian: a simulator for spiking neural networks in Python. *Frontiers in Neuroinformatics*, 2, 2008. doi: 10.3389/neuro.01.026.2009.

Fritz Haake, Joseph W Haus, and Roy Glauber. Passage-time statistics for the decay of unstable equilibrium states. *Physical Review A*, 23(6):3255, 1981.

R.H.R. Hahnloser. Emergence of neural integration in the head-direction system by visual supervision. *Neuroscience*, 120(3):877–891, 2003. ISSN 0306-4522. doi: http://dx.doi.org/10.1016/S0306-4522(03)00201-X. URL `http://www.sciencedirect.com/science/article/pii/S030645220300201X`.

Dong Han, Yuanjin Zheng, Ramamoorthy Rajkumar, Gavin Stewart Dawe, and Minkyu Je. A 0.45 v 100-channel neural-recording ic with sub-/channel consumption in 0.18 cmos. *Biomedical Circuits and Systems, IEEE Transactions on*, 7(6):735–746, 2013.

Christopher M Harris and Daniel M Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784, 1998.

R.R. Harrison. The design of integrated circuits to observe brain activity. *Proceedings of the IEEE*, 96(7):1203–1216, 2008.

R.R. Harrison and C. Charles. A low-power low-noise CMOS amplifier for neural recording applications. *IEEE Journal of Solid-State Circuits*, 38(6):958–965, June 2003. ISSN 0018-9200. doi: 10.1109/JSSC.2003.811979.

J. Hasler and B. Marr. Finding a roadmap to achieve large neuromorphic hardware systems. *Frontiers in Neuroscience*, 7(118):1–29, Sep. 2013. doi: 10.3389/fnins.2013.00118.

P. Hasler, B.A. Minch, and C. Diorio. An autozeroing floating-gate bandpass filter [in CMOS]. In *International Symposium on Circuits and Systems, (ISCAS), 1998*, volume 1, pages 131–134. IEEE, 1998. Monterey, CA, 31 May–3 June.

J. Hertz, A. Krogh, and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA, 1991.

Sean L Hill, Yun Wang, Imad Riachi, Felix Schürmann, and Henry Markram. Statistical connectivity provides a sufficient foundation for specific functional connectivity in neocortical neural microcircuits. *Proceedings of the National Academy of Sciences*, 109(42):E2885–E2894, 2012.

G.E. Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

G.E. Hinton and T.J. Sejnowski. *Unsupervised learning: foundations of neural computation*. The MIT press, 1999.

G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Submitted on 03 Jul. 2012, 2012.

Roni Hogri, Simeon A Bamford, Aryeh H Taub, Ari Magal, Paolo Del Giudice, and Matti Mintz. A neuro-inspired model-based closed-loop neuroprosthesis for the substitution of a cerebellar learning function in anesthetized rats. *Scientific reports*, 5, 2015.

J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

T. Horiuchi, T. Swindell, D. Sander, and P. Abshier. A low-power CMOS neural amplifier with amplitude measurements for spike sorting. In *International Symposium on Circuits and Systems, (ISCAS), 2004*, volume 4, pages 29–32. IEEE, May 2004.

Timothy Horiuchi, Dorielle Tucker, Kevin Boyle, and Pamela Abshire. Spike discrimination using amplitude measurements with a low-power cmos neural amplifier. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 3123–3126. IEEE, 2007.

Jonathan C Horton and Daniel L Adams. The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):837–862, 2005.

Hung-Yi Hsieh and Kea-Tiong Tang. Vlsi implementation of a bio-inspired olfactory spiking neural network. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1065–1073, July 2012. ISSN 2162-237X. doi: 10.1109/TNNLS.2012.2195329.

Feng-hsiung Hsu. Ibm's deep blue chess grandmaster chips. *IEEE Micro*, 19(2):70–81, 1999.

D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Jour. Physiol.*, 160:106–54, 1962.

K.M. Hynna and K. Boahen. Thermodynamically-equivalent silicon models of ion channels. *Neural Computation*, 19:327–350, 2007.

N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D.S. Modha. A digital neurosynaptic core using event-driven qdi circuits. In *Asynchronous Circuits and Systems (ASYNC), 2012 18th IEEE International Symposium on*, pages 25–32, May 2012. doi: 10.1109/ASYNC.2012.12.

G. Indiveri and S. Fusi. Spike-based learning in VLSI networks of integrate-and-fire neurons. In *International Symposium on Circuits and Systems, (ISCAS), 2007*, pages 3371–3374. IEEE, 2007. doi: 10.1109/ISCAS.2007.378290. URL `http://ncs.ethz.ch/pubs/pdf/Indiveri_Fusi07.pdf`.

G. Indiveri, E. Chicca, and R.J. Douglas. A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221, Jan 2006. doi: 10.1109/TNN.2005.860850. URL `http://ncs.ethz.ch/pubs/pdf/Indiveri_etal06.pdf`.

G. Indiveri, B. Linares-Barranco, T.J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5:1–23, 2011. ISSN 1662-453X. doi: 10.3389/fnins.2011.00073. URL `http://www.frontiersin.org/Neuromorphic_Engineering/10.3389/fnins.2011.00073/abstract`.

G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis. Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology*, 24(38):384010, 2013. doi: 10.1088/0957-4484/24/38/384010. URL `http://stacks.iop.org/0957-4484/24/i=38/a=384010`.

Giacomo Indiveri and Shih-Chii Liu. Memory and information processing in neuromorphic systems. *arXiv preprint arXiv:1506.03264*, 2015.

M. Itō. *The cerebellum and neural control*. Raven Press, 1984.

E.M. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6):1569–1572, 2003.

E.M. Izhikevich. *Dynamical systems in neuroscience: The geometry of excitability and bursting*. The MIT press, 2006.

E.M. Izhikevich and G. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Science*, 105:3593–3598, 2008. doi: 10.1073/pnas.0712231105.

MR Jarvis and PP Mitra. Sampling properties of the spectrum and coherency of sequences of action potentials. *Neural Computation*, 13(4):717–749, 2001.

X. Jin, M Lujan, L.A. Plana, S. Davies, S. Temple, and S. Furber. Modeling spiking neural networks on SpiNNaker. *Computing in Science & Engineering*, 12(5):91–97, September-October 2010.

Matthias Kaschube, Michael Schnabel, Siegrid Löwel, David M Coppola, Leonard E White, and Fred Wolf. Universality in the evolution of orientation columns in the visual cortex. *science*, 330(6007): 1113–1116, 2010.

B Katz and Ri Miledi. The statistical nature of the acetylcholine potential and its molecular components. *The Journal of physiology*, 224(3):665, 1972.

Matthew B Kennel, Reggie Brown, and Henry DI Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, 45(6):3403, 1992.

P. Kmon and P. Grybos. Energy efficient low-noise multichannel neural amplifier in submicron cmos process. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(7):1764–1775, July 2013. ISSN 1549-8328. doi: 10.1109/TCSI.2012.2230504.

Luis F Lago-Fernández and Gustavo Deco. A model of binocular rivalry based on competition in it. *Neurocomputing*, 44:503–507, 2002.

John Lazzaro, John Wawrzynek, Misha Mahowald, Massimo Sivilotti, and Dave Gillespie. Silicon auditory processors as computer peripherals. *Neural Networks, IEEE Transactions on*, 4(3):523–528, 1993.

Q.V. Le, M.A. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, and A.Y. Ng. Building high-level features using large scale unsupervised learning. Last revised 12 Jul 2012 (v5), 2012.

Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Y. Le Cun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, Mar 2015. doi: 10.1038/nature14539.

P. Lichtsteiner, C. Posch, and T. Delbruck. An 128x128 120dB 15μs-latency temporal contrast vision sensor. *IEEE J. Solid State Circuits*, 43(2):566–576, 2008.

Konstantin Likharev. Electronics below 10 mm. *Nano and Giga Challenges in Microelectronics*, page 27, 2003.

B. Linares-Barranco and T. Serrano-Gotarredona. On the design and characterization of femtoampere current-mode circuits. *IEEE J. Solid-State Circuits*, 38(8):1353–1363, August 2003.

J. Lisman and N. Spruston. Postsynaptic depolarization requirements for ltp and ltd: a critique of spike timing-dependent plasticity. *Nature Neuroscience*, 8(7):839–841, Jul 2005.

J. Lisman and N. Spruston. Questions about stdp as a general model of synaptic plasticity. *Frontiers in Synaptic Neuroscience*, 2(140):1–3, 2010. doi: 10.3389/fnsyn.2010.00140.

John E Lisman, Sridhar Raghavachari, and Richard W Tsien. The sequence of events that underlie quantal transmission at central glutamatergic synapses. *Nature Reviews Neuroscience*, 8(8):597–609, 2007.

S.-C. Liu. Analog VLSI circuits for short-term dynamic synapses. *European Journal on Applied Signal Processing*, 7:1–9, 2003.

S.-C. Liu and T. Delbruck. Neuromorphic sensory systems. *Current Opinion in Neurobiology*, 20(3): 288–295, 2010. doi: 10.1016/j.conb.2010.03.007.

S.-C. Liu, J. Kramer, G. Indiveri, T. Delbruck, and R.J. Douglas. *Analog VLSI:Circuits and Principles*. MIT Press, 2002. URL `http://ncs.ethz.ch/pubs/pdf/Liu_etal02b.pdf`.

Shih-Chii Liu. *Event-Based Neuromorphic Systems*. John Wiley & Sons, 2015.

P. Livi and G. Indiveri. A current-mode conductance-based silicon neuron for address-event neuromorphic systems. In *International Symposium on Circuits and Systems, (ISCAS), 2009*, pages 2898–2901. IEEE, May 2009. doi: 10.1109/ISCAS.2009.5118408. URL `http://ncs.ethz.ch/pubs/pdf/Livi_Indiveri09.pdf`.

Cheng-Hung Lo and Shi-Yu Huang. Ppn based 10t sram cell for low-leakage and resilient subthreshold operation. *Solid-State Circuits, IEEE Journal of*, 46(3):695–704, 2011.

C.M. Lopez, A. Andrei, S. Mitra, M. Welkenhuysen, W. Eberle, C. Bartic, R. Puers, R.F. Yazicioglu, and G.G.E. Gielen. An implantable 455-active-electrode 52-channel CMOS neural probe. *IEEE Journal of Solid-State Circuits*, 49(1):248–261, Jan 2014. ISSN 0018-9200. doi: 10.1109/JSSC.2013.2284347.

Theodore M. Wong, Robert Preissl, Pallab Datta, Myron Flicker, Raghavendra Singh, Steven K. Esser, Emmet McQuinn, Rathinakumar Appuswamy, William P. Risk, Horst D. Simon, and Dharmendra S. Modha. Ten power 14, 2013.

W. Maass. On the computational power of winner-take-all. *Neural Computation*, 12(11):2519–2535, 2000.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

C.A. Mack. Fifty years of moore's law. *Semiconductor Manufacturing, IEEE Transactions on*, 24(2): 202–207, 2011.

Donald M MacKay and Warren S McCulloch. The limiting information capacity of a neuronal link. *The bulletin of mathematical biophysics*, 14(2):127–135, 1952.

L.P. Maguire, T. M. McGinnity, B. Glackin, A. Ghani, A. Belatreche, and J. Harkin. Challenges for large-scale implementations of spiking neural networks on FPGAs. *Neurocomputing*, 71(1):13–29, 2007.

M. Mahowald and R.J. Douglas. A silicon neuron. *Nature*, 354:515–518, 1991.

S. Mandal and R. Sarpeshkar. A bidirectional wireless link for neural prostheses that minimizes implanted power consumption. In *Biomedical Circuits and Systems Conference, (BioCAS), 2007*, pages 45–48. IEEE, Nov. 2007. doi: 10.1109/BIOCAS.2007.4463305.

R. Manohar. Reconfigurable asynchronous logic. In *Custom Integrated Circuits Conference*, pages 13–20. IEEE, 2006.

John Markoff. Computer wins on 'jeopardy!': trivial, it's not. *New York Times*, 16, 2011.

H. Markram. The human brain project. *Scientific American*, 306(6):50–55, 2012.

H. Markram, W. Gerstner, and P.J. Sjöström. Spike-timing-dependent plasticity: a comprehensive overview. *Frontiers in Synaptic Neuroscience*, 4(2):1–3, 2012. doi: 10.3389/fnsyn.2012.00002.

Henry Markram. The blue brain project. *Nature Reviews Neuroscience*, 7(2):153–160, 2006.

Alain J Martin, Andrew Lines, Rajit Manohar, Mika Nystroem, Paul Penzes, Robert Southworth, and Uri Cummings. The design of an asynchronous mips r3000 microprocessor. In *arvlsi*, page 164. IEEE, 1997.

Norbert Marwan, M Carmen Romano, Marco Thiel, and Jürgen Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5):237–329, 2007.

Massimo Mascaro and Daniel J Amit. Effective neural response function for collective population states. *Network: Computation in Neural Systems*, 10(4):351–373, 1999. doi: 10.1088/0954-898X_10_4_305.

T. Masquelier, R. Guyonneau, and S.J. Thorpe. Competitive STDP-based spike pattern learning. *Neural Computation*, 21(5):1259–1276, 2009. doi: 10.1162/neco.2008.06-08-804.

T.M. Massoud and T.K. Horiuchi. A neuromorphic head direction cell system. In *International Symposium on Circuits and Systems, (ISCAS), 2009*. IEEE, 2009.

Emmett McQuinn, Pallab Datta, Myron D Flickner, William P Risk, and Dharmendra S Modha. Connectivity of a cognitive computer based on the macaque brain. *Science*, 339(6119):513–513, 2013.

C. Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–36, 1990.

C.A. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA, 1989.

P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D.S. Modha. A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm. In *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, pages 1–4, Sept. 2011. doi: 10.1109/CICC.2011.6055294.

P. Merolla, J. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B.L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S.K. Esser, R. Appuswamy, B. Taba, A. Amir, M.D. Flickner, W.P. Risk, R. Manohar, and D.S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014. doi: 10.1126/science.1254642.

P.A. Merolla, J.V. Arthur, B.E. Shi, and K.A. Boahen. Expandable networks for neuromorphic chips. *IEEE Transactions on Circuits and Systems I*, 54(2):301–311, Feb. 2007.

S. Mitra, G. Indiveri, and S. Fusi. Learning to classify complex patterns using a VLSI network of spiking neurons. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 1009–1016, Cambridge (MA), 2008. MIT Press. URL `http://ncs.ethz.ch/pubs/pdf/Mitra_etal08.pdf`.

S. Mitra, S. Fusi, and G. Indiveri. Real-time classification of complex patterns using spike-based learning in neuromorphic VLSI. *Biomedical Circuits and Systems, IEEE Transactions on*, 3(1):32–42, Feb. 2009. doi: 10.1109/TBCAS.2008.2005781. URL `http://ncs.ethz.ch/pubs/pdf/Mitra_etal09.pdf`.

Dharmendra S Modha, Rajagopal Ananthanarayanan, Steven K Esser, Anthony Ndirango, Anthony J Sherbondy, and Raghavendra Singh. Cognitive computing. *Communications of the ACM*, 54(8):62–71, 2011.

A. Mohamed, G.E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):14–22, Jan 2012. doi: 10.1109/TASL.2011.2109382.

S. Moradi and G. Indiveri. A VLSI network of spiking neurons with an asynchronous static random access memory. In *Biomedical Circuits and Systems Conference (BioCAS), 2011*, pages 277–280. IEEE, 2011. doi: 10.1109/BioCAS.2011.6107781. URL `http://ncs.ethz.ch/pubs/pdf/Moradi_Indiveri11.pdf`.

S. Moradi and G. Indiveri. An event-based neural network architecture with an asynchronous programmable synaptic memory. *Biomedical Circuits and Systems, IEEE Transactions on*, 8(1):98–107, February 2014. doi: 10.1109/TBCAS.2013.2255873. URL `http://ncs.ethz.ch/pubs/pdf/Moradi_Indiveri14.pdf`.

H. Mostafa, F. Corradi, M. Osswald, and G. Indiveri. Automated synthesis of asynchronous event-based interfaces for neuromorphic systems. In *Circuit Theory and Design, (ECCTD) 2013 European Conference on*, pages 1–4. IEEE, 2013. doi: 10.1109/ECCTD.2013.6662213. URL `http://ncs.ethz.ch/pubs/pdf/Mostafa_etal13.pdf`.

H. Mostafa, F. Corradi, F. Stefanini, and G. Indiveri. A hybrid analog/digital spike-timing dependent plasticity learning circuit for neuromorphic VLSI multi-neuron architectures. In *International Symposium on Circuits and Systems, (ISCAS), 2014*, pages 854–857. IEEE, 2014. URL `http://ncs.ethz.ch/pubs/pdf/Mostafa_etal14.pdf`.

VB Mountcastle, AL Berman, and PW Davies. Topographic organization and modality representation in first somatic area of cat's cerebral cortex by method of single unit analysis. *Am. J. Physiol*, 183:464, 1955.

Lorenz K Muller and Giacomo Indiveri. Rounding methods for neural networks with low resolution synaptic weights. *arXiv preprint arXiv:1504.05767*, 2015.

E. Neftci, J. Binas, U. Rutishauser, E. Chicca, G. Indiveri, and R. Douglas. Synthesizing cognition in neuromorphic electronic systems. *Proceedings of the National Academy of Sciences*, 110(37):E3468–E3476, 2013. doi: 10.1073/pnas.1212083110. URL `http://ncs.ethz.ch/pubs/pdf/Neftci_etal13.pdf`.

D. Neil and S.-C. Liu. Minitaur, an event-driven FPGA-based spiking network accelerator. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, PP(99):1–1, October 2014. doi: 10.1109/TVLSI.2013.2294916.

B. Nessler, M. Pfeiffer, and W. Maass. STDP enables spiking neurons to detect hidden causes of their inputs. In Y. Bengio, D. Schuurmans, J. Lafferty, C.K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1357–1365, 2009.

Ernst Niebur. Electrophysiological correlates of synchronous neural activity and attention: A short review. *Biosystems*, 67(1):157–166, 2002.

Lars S Nielsen, Cees Niessen, Jens Sparso, and Kees Van Berkel. Low-power operation using self-timed circuits and adaptive scaling of the supply voltage. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 2(4):391–397, 1994.

A.V. Nurmikko, J.P. Donoghue, L.R. Hochberg, W.R. Patterson, Y.K. Song, C.W. Bull, D.A. Borton, F. Laiwalla, S. Park, Y. Ming, and J. Aceros. Listening to brain microcircuits for interfacing with externalworld, progress in wireless implantable microelectronic neuroengineering devices. *Proceedings of the IEEE*, 98(3):375–388, 2010. doi: http://dx.doi.org/10.1109/JPROC.2009.2038949.

P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, 7(178), 2013. ISSN 1662-453X. doi: 10.3389/fnins.2013.00178. URL `http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2013.00178/abstract`.

B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by vi? *Vision research*, 37(23):3311–3326, 1997.

E. Painkras, L.A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D.R. Lester, A.D. Brown, and S.B. Furber. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):–, August 2013. ISSN 0018-9200. doi: 10.1109/JSSC.2013.2259038.

G. Passetti, F. Corradi, M. Raglianti, D. Zambrano, C. Laschi, and G. Indiveri. Implementation of a neuromorphic vestibular sensor with analog VLSI neurons. In *Biomedical Circuits and Systems Conference, (BioCAS), 2013*, pages 174–177. IEEE, Oct. 2013. doi: 10.1109/BioCAS.2013.6679667. URL `http://ncs.ethz.ch/pubs/pdf/Passetti_etal13.pdf`.

Rodrigo Perin, Thomas K Berger, and Henry Markram. A synaptic organizing principle for cortical neuronal groups. *Proceedings of the National Academy of Sciences*, 108(13):5419–5424, 2011.

M.A. Petrovici, B. Vogginger, P. Müller, O. Breitwieser, M. Lundqvist, L. Muller, M. Ehrlich, A. Destexhe, A. Lansner, R. Schüffny, J. Schemmel, and K. Meier. Characterization and compensation of network-level anomalies in mixed-signal neuromorphic modeling platforms. *PloS one*, 9(10):e108590, 2014. doi: 10.1371/journal.pone.0108590.

T. Pfeil, T. C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier. Is a 4-bit synaptic weight resolution enough? - constraints on enabling spike-timing dependent plasticity in neuromorphic hardware. *Frontiers in Neuroscience*, 6, 2012. ISSN 1662-453X. doi: 10.3389/fnins.2012.00090. URL `http://www.frontiersin.org/Journal/Abstract.aspx?s=755&name=neuromorphic_engineering&ART_DOI=10.3389/fnins.2012.00090`.

Robert Preissl, Theodore M Wong, Pallab Datta, Myron Flickner, Raghavendra Singh, Steven K Esser, William P Risk, Horst D Simon, and Dharmendra S Modha. Compass: A scalable simulator for an architecture for cognitive computing. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, page 54. IEEE Computer Society Press, 2012.

Mirko Prezioso, Farnood Merrikh-Bayat, BD Hoskins, GC Adam, Konstantin K Likharev, and Dmitri B Strukov. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature*, 521(7550):61–64, 2015.

N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri. A re-configurable on-line learning spiking neuromorphic processor. *Frontiers in Neuroscience*, 2015.

S. Ramakrishnan, R. Wunderlich, and P. Hasler. Neuron array with plastic synapses and programmable dendrites. In *Biomedical Circuits and Systems Conference, (BioCAS), 2012*, pages 400–403. IEEE, Nov. 2012. doi: 10.1109/BioCAS.2012.6418412.

C. Rasche and R. Hahnloser. Silicon synaptic depression. *Biological Cybernetics*, 84(1):57–62, 2001.

A.D. Rast, S. Yang, M. Khan, and S.B. Furber. Virtual synaptic interconnect using an asynchronous network-on-chip. In *International Joint Conference on Neural Networks, IJCNN 2008*, pages 2727–2734. IEEE, 2008.

F. Rieke. *Spikes: Exploring the neural code*. The MIT Press, 1997.

M. Rigotti, D.D. Ben Dayan Rubin, S.E. Morrison, C.D. Salzman, and S. Fusi. Attractor concretion as a mechanism for the formation of context representations. *NeuroImage*, 52(3):833–847, 2010. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2010.01.047. URL `http://www.sciencedirect.com/science/article/pii/S1053811910000698`.

E.T. Rolls and G. Deco. *Computational Neuroscience of Vision*. Oxford University Press, Oxford, 2002.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review,*, 65(6):386–408, Nov 1958. doi: 10.1037/h0042519.

Michael T Rosenstein, James J Collins, and Carlo J De Luca. A practical method for calculating largest lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, 65(1):117–134, 1993.

John Ross, David Burr, and Concetta Morrone. Suppression of the magnocellular pathway during saccades. *Behavioural brain research*, 80(1):1–8, 1996.

G. Rovere, Q. Ning, C. Bartolozzi, and G. Indiveri. Ultra low leakage synaptic scaling circuits for implementing homeostatic plasticity in neuromorphic architectures. In *International Symposium on Circuits and Systems, (ISCAS), 2014*, pages 2073–2076. IEEE, 2014. URL `http://ncs.ethz.ch/pubs/pdf/Rovere_etal14.pdf`.

David E Rumelhart, Geoffrey E Hintont, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

S.G. Sadeghi, L.B. Minor, and K.E. Cullen. Response of vestibular-nerve afferents to active and passive rotations under normal conditions and after unilateral labyrinthectomy. *Journal of neurophysiology*, 97 (2):1503–1514, November 2006.

S.G. Sadeghi, M.J. Chacron, M.C. Taylor, and K.E. Cullen. Neural variability, detection thresholds, and information transmission in the vestibular system. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(4):771–781, January 2007.

R. Sarpeshkar, R.F. Lyon, and C.A. Mead. An analog VLSI cochlea with new transconductance amplifiers and nonlinear gain control. In *Proc. IEEE Int. Symp. on Circuits and Systems*, volume 3, pages 292–296. IEEE, May 1996.

R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press (MA), 2012.

J. Schemmel, D. Bruderle, A. Grubl, M. Hock, K. Meier, and S. Millner. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1947–1950. IEEE, 2010.

M. Schmuker, T. Pfeil, and M.P. Nawrot. A neuromorphic network for generic multivariate data classification. *Proceedings of the National Academy of Sciences*, 111(6):2081–2086, 2014. doi: 10.1073/pnas.1303053111. URL `http://www.pnas.org/content/111/6/2081.abstract`.

W. Senn. Beyond spike timing: the role of nonlinear plasticity and unreliable synapses. *Biol. Cybern.*, 87: 344–355, 2002.

W. Senn and S. Fusi. Learning Only When Necessary: Better Memories of Correlated Patterns in Networks with Bounded Synapses. *Neural Computation*, 17(10):2106–2138, 2005. URL `http://neco.mitpress.org/cgi/content/abstract/17/10/2106`.

J. Seo, B. Brezzo, Y. Liu, B.D. Parker, S.K. Esser, R.K. Montoye, B. Rajendran, J. Tierno, L. Chang, and D.S. Modha. A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, pages 1–4. IEEE, 2011.

R. Serrano-Gotarredona, T. Serrano-Gotarredona, A. Acosta-Jimenez, C. Serrano-Gotarredona, J.A. Perez-Carrasco, A. Linares-Barranco, G. Jimenez-Moreno, A. Civit-Ballcels, and B. Linares-Barranco. On real-time aer 2d convolutions hardware for neuromorphic spike based cortical processing. *IEEE Transactions on Neural Networks*, 19(7):1196–1219, July 2008.

R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Häfliger, G. Jimenez-Moreno, A. Civit-Ballcels, T. Serrano-Gotarredona, A.J. Acosta-Jiménez, and B. Linares-Barranco. CAVIAR: A 45k neuron, 5M synapse, 12G connects/s aer hardware sensory–processing– learning–actuating system for high-speed visual object recognition and tracking. *IEEE Transactions on Neural Networks*, 20(9):1417–1438, September 2009. doi: 10.1109/TNN.2009.2023653.

S. Sheik, E. Chicca, and G. Indiveri. Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays. In *International Joint Conference on Neural Networks, IJCNN 2012*, pages 1940–1945. IEEE, 2012a. doi: 10.1109/IJCNN.2012.6252636. URL `http://ncs.ethz.ch/pubs/pdf/Sheik_etal12b.pdf`.

S. Sheik, M. Coath, G. Indiveri, S.L. Denham, T. Wennekers, and E. Chicca. Emergent auditory feature tuning in a real-time neuromorphic VLSI system. *Frontiers in Neuroscience*, 6(17), 2012b. doi: 10.3389/fnins.2012.00017. URL `http://ncs.ethz.ch/pubs/pdf/Sheik_etal12.pdf`.

S. Sheik, M. Pfeiffer, F. Stefanini, and G. Indiveri. Spatio-temporal spike pattern classification in neuromorphic systems. In *Biomimetic and Biohybrid Systems*, pages 262–273. Springer, 2013. doi: 10.1007/978-3-642-39802-5_23. URL `http://ncs.ethz.ch/pubs/pdf/Sheik_etal13.pdf`.

T Shibata and Stefan Schaal. Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks. *Neural Networks*, January 2001.

Marina Skurichina and Robert PW Duin. Bagging, boosting and the random subspace method for linear classifiers. *Pattern Analysis & Applications*, 5(2):121–135, 2002.

Charles E Smith and Jay M Goldberg. A stochastic afterhyperpolarization model of repetitive activity in vestibular afferents. *Biological cybernetics*, 54(1):41–51, 1986.

J. Sparsø. Asynchronous circuit design - a tutorial. In *Chapters 1-8 in "Principles of asynchronous circuit design - A systems Perspective"*, pages 1–152. Kluwer Academic Publishers, Boston / Dordrecht / London, dec 2001. URL `http://www2.imm.dtu.dk/pubdb/p.php?855`.

F. Stefanini, S. Sheik, E. Neftci, and G. Indiveri. Pyncs: a microkernel for high-level definition and configuration of neuromorphic electronic systems. *Frontiers in Neuroinformatics*, 8(73), 2014. doi: 10.3389/fninf.2014.00073. URL `http://ncs.ethz.ch/pubs/pdf/Stefanini_etal14.pdf`.

P.N. Steinmetz, A. Roy, P Fitzgerald, S.S. Hsiao, K.O. Johnson, and E. Niebur. Attention modulates synchronized neuronal firing in primate somatosensory cortex. *Nature*, 404:187–190, 2000.

I.H. Stevenson and K.P. Kording. Dynamic causal models of neural system dynamics: current state and future extensions. *Nature Neuroscience*, 14(2):139–142, 2011. doi: 10.1038/nn.2731.

Evangelos Stromatias, Daniel Neil, Michael Pfeiffer, Francesco Galluppi, Steve B Furber, and S Liu. Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Name: Frontiers in Neuroscience*, 9:222, 2015.

Floris Takens. *Detecting strange attractors in turbulence*. Springer, 1981.

Wei Tang, Ahmad Osman, Dongsoo Kim, Brian Goldstein, Chenxi Huang, Berin Martini, Vincent A Pieribone, and Eugenio Culurciello. Continuous time level crossing sampling adc for bio-potential recording systems. *IEEE transactions on circuits and systems. I, Regular papers: a publication of the IEEE Circuits and Systems Society*, 60(6):1407, 2013.

Neil P McAngus Todd, Sally M Rosengren, and James G Colebatch. Tuning and sensitivity of the human vestibular system to low-frequency vibration. *Neuroscience letters*, 444(1):36–41, 2008.

Özgür Türel and Konstantin Likharev. Crossnets: Possible neuromorphic networks based on nanoscale components. *International journal of circuit theory and applications*, 31(1):37–53, 2003.

Kees Van Berkel, Ronan Burgess, Joep LW Kessels, Ad Peeters, Marly Roncken, and Frits Schalij. A fully asynchronous low-power error corrector for the dcc player. *Solid-State Circuits, IEEE Journal of*, 29(12):1429–1439, 1994.

A. van Schaik and R. Meddis. Analog very large-scale integrated (VLSI) implementation of a model of amplitude-modulation sensitivity in the auditory brainstem. *The Journal of the Acoustical Society of America*, 105:811, 1999.

A. van Schaik, C. Jin, T.J Hamilton, S. Mihalas, and E. Niebur. A log-domain implementation of the Mihalas-Niebur neuron model. In *International Symposium on Circuits and Systems, (ISCAS), 2010*, pages 4249–4252, Paris, France, 2010. IEEE.

V. Vapnik. *The nature of statical learning theory*. Springer Verlag, 1995.

R. Wang, G. Cohen, K.M. Stiefel, T.J. Hamilton, J. Tapson, and A. van Schaik. An FPGA implementation of a polychronous spiking neural network with delay adaptation. *Frontiers in neuroscience*, 7, 2013.

Runchun Wang, T.J. Hamilton, J. Tapson, and A. van Schaik. A compact neural core for digital implementation of the neural engineering framework. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 548–551, Oct 2014. doi: 10.1109/BioCAS.2014.6981784.

X. Wang. Synaptic basis of cortical persistent activity: the importance of NMDA receptors to working memory. *Journal of Neuroscience*, 19:9587–9603, November 1999.

Xiao-Jing Wang. Neural dynamics and circuit mechanisms of decision-making. *Current Opinion in Neurobiology*, 2012.

X.J. Wang. Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, 36(5): 955–968, 2002.

X.J. Wang. Decision making in recurrent neuronal circuits. *Neuron*, 60:215–234, Oct 2008.

W. Wattanapanitch and R. Sarpeshkar. A low-power 32-channel digitally programmable neural recording integrated circuit. *IEEE Transactions on Biomedical Circuits and Systems*, 5(6):592–602, Dec 2011. ISSN 1932-4545. doi: 10.1109/TBCAS.2011.2163404.

By Kensall D Wise, Amir M Sodagar, Ying Yao, Mayurachat Ning Gulari, Gayatri E Perlin, and Khalil Najafi. Microelectrodes, microelectronics, and implantable neural microsystems. *Proceedings of the IEEE*, 96(7):1184–1202, 2008.

Gayle M Wittenberg, Megan R Sullivan, and Joe Z Tsien. Synaptic reentry reinforcement based network model for long-term memory consolidation. *Hippocampus*, 12(5):637–647, 2002.

Kong-Fatt Wong and Xiao-Jing Wang. A recurrent network mechanism of time integration in perceptual decisions. *The Journal of neuroscience*, 26(4):1314–1328, 2006.

R. Wood, A. McGlashan, J. Yatulis, P. Mascher, and I. Bruce. Digital implementation of a neural network for imaging. In *Photonics North 2012*, pages 84121H–84121H. International Society for Optics and Photonics, 2012.

X. Xie, R.H.R. Hahnloser, and H.S. Seung. Double-ring network model of the head-direction system. *Physical Review E*, 66(4):04192, 2002.

P. Xu, T.-K. Horiuchi, A. Sarje, and P. Abshire. Stochastic synapse with short-term depression for silicon neurons. In *Biomedical Circuits and Systems Conference, 2007. BIOCAS 2007. IEEE*, pages 99–102, 2007. doi: 10.1109/BIOCAS.2007.4463318.

Sergei B Yakushin, Theodore Raphan, and Bernard Cohen. Spatial properties of central vestibular neurons. *Journal of neurophysiology*, 95(1):464–478, 2006.

T. Yu, J. Park, S. Joshi, C. Maier, and G. Cauwenberghs. 65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing. In *Biomedical Circuits and Systems Conference, (BioCAS), 2012*, pages 21–24. IEEE, Nov. 2012. doi: 10.1109/BioCAS.2012.6418479.

K.A. Zaghloul and K. Boahen. Optic nerve signals in a neuromorphic chip: Parts 1&2. *Biomedical Circuits and Systems, IEEE Transactions on*, 51:657–675, 2004.

Jie Zhang, Srinjoy Mitra, Yuanming Suo, Andrew Cheng, Tao Xiong, Frederic Michon, Marleen Welkenhuysen, Fabian Kloosterman, Peter S Chin, Steven Hsiao, et al. A closed-loop compressive-sensing-based neural recording system. *Journal of neural engineering*, 12(3):036005, 2015.

K. Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *The journal of neuroscience*, 16(6):2112–2126, 1996.

WS Zhao, Guillaume Agnus, Vincent Derycke, A Filoramo, JP Bourgoin, and C Gamrat. Nanotube devices based crossbar architecture: toward neuromorphic computing. *Nanotechnology*, 21(17):175202, 2010.

Zusammenfassung

Die Integration von Gehirn-ähnlichen Fähigkeiten in elektronischen Systemen erfordert ein Verständnis der organisatorischen Grundsätze von Neuronen und Synapsen in Neurobiologischen Systemen.

Nervensysteme bearbeiten Information auf eine sehr unterschiedliche Weise als heutige IT- und Computer-Systeme. Sie benutzen Milliarden langsame, inhomogene und in der Genauigkeit beschränkte Verarbeitungseinheiten (Neuronen und Synapsen), und trotzdem übertreffen sie die Leistung von herkömmlichen Computer-Systemen bei einer Reihe von Aufgaben, zb. Spracherkennung, Spracherzeugung, Motorische Kontrolle, und die Integration mehrerer Sensoren.

Neuromorphische Ansätze haben sich als vielversprechende Alternativen zu von Neumann Computer-Architekturen hervorgehoben. Diese Ansätze versuchen nicht die Rechengeschwindigkeit von normalen, algoritmischen Computer-Systemen zu erhöhen, sondern nehmen ihre Inspiration aus der Leistungsfähigkeit und der Wiederstandsfähigkeit von biologischen Nervensystemen, und versuchen diese Prinzipien in Very Large Scale Integration (VLSI) Systemen nachzubilden.

Die Herausforderung steht im Verständnis der Komplexität von neurobiologischen Systemen zusammen mit der Integration von experimentellen Erkentnissen auf verschiedenen Ebenen; die, von intelligentem Verhalten bis zu biochemischen Prozessen, sich über weit mehr als sechs Grössenordnungen in Raum und Zeit aufspannen (von Nanometer bis Millimeter, von Mikrosekunden bis Sekunden).

Die Hauptfragen die ich versucht habe zu beantworten sind: wie können wir von Neuronen, zu Rechenfunktionen, zu Verhaltenssysteme gelangen? And vor allem, gibt es irgendeine Neurale Schaltung die man als Wiederverwendbare Komponente in Neuralen Systemen benützen könnte? Welche sind die rechnerischen Grundkonstrukte die für Gedächtnis, assoziatives Lernen, und Entscheidungsvermögen nötig sind? Wie können wir diese Grundkonstrukte in mikroelektronische Schaltungen verwirklichen?

Um diese Fragen zu beantworten, habe ich (zusammen mit meinen Kollegen) spezielle gemischte-Signale analog/digital VLSI Hardware-Architekturen entwickelt. Diese Architekturen eignen sich ideal als Platform für die Nachbildung von Neuronen, da sie kompakte und Energieeffiziente Implementationen von massiv parallelen verteilten Spike-basierten Systemen sind. Namentlich, habe ich mehrere biologisch Inspirierte Schaltungen entwickelt und hergestellt, die sich so verhalten, wie ihre neurobiologischen Gegenstücke. Ich nahm mir die Herausforderung an, verteilte und programmierbare Berechnungen mit gestörten und heterogenen analogen Schaltungen zu realisieren, die in Netzwerke von asynchronen Spiking-Neuronen organisiert waren. Dazu studierte ich die Stabilität von Attraktoren-Zustände, die aus dicht zusammengedrängten, wiederkehrender Netzwerke von Spiking-Neuronen

hervortreten. Ich zeigte wie diese mikroelektrischen neurale Schaltungen einige der Verhalten, die in neurobiologischen Netzwerke gesehen wurden, stabil wiedergegeben können. Ich zeigte auch, mittels Experimenten, dass es möglich ist, mit VLSI multi-Neuronen Systeme, beliebige matematische Rechnungen durchzuführen.

Um weiter zu erforschen wie echte neurale Systeme rechnen, habe ich ein neuromorphisches System realisiert, dass von neuralem Gewebe Signale aufzeichnen kann. Dieses System ist eine Ereignis-basierte Gehirn-Machine-Schnittstelle, die asynchrone Logik benutzt, um die Signale vom neuralen Gewebe sammeln und übertragen zu können. Das Hauptziel unseres Entwurfes ist nicht eine wirklichkeitsgetreue Nachbildung von Aktionspotentiale, sondern effizientes und schnelles Rechnen auf Grund weniger und komprimierter Daten. Diese Forschung eröffnet neue zukünftliche Möglichkeiten im Bereich der Gehirn-Machine-Schnittstellen, vor allem dort wo keine genaue Spike-Aufteilung nötig ist, sondern eine schnelle Übertragung und Komprimierung eines hoch-dimensionalen Signals.Die Resultate dieser Diplomarbeit zeigen einen Weg um optimale Entwürfe einer neuen Generation von Rechnertechnologien zu ermöglichen, die auf hybride neuromorphische analog/digital VLSI Schaltungen basieren, die in wiederverwendbaren, kleinen Netzwerken aufgebaut sind.